ABC80

BASIC: Akielen ABC

otava

Anders Andersson
Arne Kullbjer
Jan Lundgren
Sören Thornell

BASIC-kielen ABC

PKA
61.39

Kotkan
Kaupunginkirjasto

HELSINGISSÄ KUSTANNUSOSAKEYHTIÖ OTAVA

4

Suomentajat: Kauko Orrainen ja Esko Peuhkuri

Piirrokset: Ulf Nilsson, Marianne Lundin
Valokuvat: Kari Lehto
2.—4. painos
© 1979 Anders Andersson, Arne Kullbjer, Jan Lundgren ja Sören Thornell
© 1980 suomenkielinen laitos Kauko Orrainen, Esko Peuhkuri ja Kustannusosakeyhtiö Otava

Kustannusosakeyhtiö Otavan painolaitokset Keuruu 1983

ISBN 951-1-06080-5

Esipuhe

Elektroniikan nopea kehitys on tehnyt mahdolliseksi tehokkaiden tietokoneiden käytön monissa uusissa sovelluksissa. Tänään voi yksityishenkilökin ostaa tietokoneen väritelevision hinnalla. Tällaiset tietokoneet ovat lisäksi niin tehokkaita, että vastaavat koneet maksoivat kymmenisen vuotta sitten noin miljoona markkaa. Niinpä on muodostunut kokonaan uusi jatkuvasti laajeneva tietokoneen käyttäjälukka, joka vaatii perustietoja uuden tekniikan kohtaamiseen. Jotta tietokoneen käyttäjä voisi hyödyntää laitteistoa oikealla tavalla ja saada samalla myös muuntelumahdollisuuksia valmiisiin ohjelmiin on tärkeä oppia tietokoneen ohjelmoinnin perustiedot. Tämä kirja antaa nämä tiedot ja taidot.

Tätä kirjaa on helpointa käyttää ABC 80 tietokoneen kanssa, mutta se soveltuu opiskeluun myös muiden nykyaikaisten mikrotietokoneiden kanssa niiden perustoimintojen osalta. Sitä voidaan niinikään vaivattomasti käyttää itseopiskeluun. Kirjan perusteellisuus mahdollistaa oikealla tavalla painottuneen opiskelun lukioissa, ammatillisissa oppilaitoksissa sekä erilaisten yritysten sisäisessä koulutuksessa.

Tämän kirjan hyväksikäyttö ei edellytä minkäänlaisia alkutietoja tietokoneista eikä niiden ohjelmoinnista.

Kirjan seitsemän ensimmäistä lukua käsittelevät yleisiä tietokoneiden käyttöalueita, kun taas seitsemän viimeistä lukua kuvaavat erikoistuneempia sovellusalueita kuten sanojen käsittelyä ja graafista esitystä.

Kiitämme kaikkia tämän kirjan ilmestymistä auttaneita henkilöitä, joista mainittakoon SCANDIA METRIC'in, LUXOR'in ja DIAB'in yhteistyöhenkilö.

Linköping, helmikuussa 1979

Tekijät

Suomenkielisen laitoksen esipuhe

Ruotsalainen alkuteos ABC om BASIC herätti heti ilmestyessään suurta kiinnostusta myös oman maamme opetus- ja koulutustehtävistä vastaavissa piireissä. Kun kirja sittemmin sai erinomaisen vastaanoton ATK-opetuksen aloittaneissa sekä sitä kokeilevissa kouluissa ja oppilaitoksissa, päätettiin kirja toimittaa myös suomenkielisenä laitoksena. Valinta suoran ja sovelletun kääntämisen välillä testattiin paitsi edellämainituissa oppilaitoksissa, myös kesällä 1980 pidetyillä koulutushallituksen tukemilla MAOL:n ja Finn METRIC Oy:n yhteistyönä järjestämillä opettajien jatkokoulutuskursseilla. Aineisto todettiin sekä lähtökohdiltaan että toteutustavaltaan poikkeuksellisen onnistuneeksi, joka ei suuria muodonmuutoksia kaivannut. Suomenkielisen laitoksen tekijät kiittävät saamastaan tuesta ja arvokkaista vihjeistä kaikkia mukanaolleita ja erityisesti Finn METRIC Oy:tä ja Kustannusosakeyhtiö Otavaa saamastaan sekä henkisestä että laitetuesta kirjan toimittamisessa suomenkielisen koulutuksen tarpeisiin.

Espoossa, lokakuussa 1980

Esko Peuhkuri
Kauko Orrainen

Sisällysluettelo

KAPPALE 1. Tietokone otetaan käyttöön 1
KAPPALE 2. Tietokone älykkäänä kirjoituskoneena 7
KAPPALE 3. Tietokone etsii tietoja 19
KAPPALE 4. Tietokone suorittaa laskuja 31
KAPPALE 5. Tietokone tekee päätöksiä 43
KAPPALE 6. Tietokone tallentaa ohjelmia 61
KAPPALE 7. Lohkokaaviot ohjelmoinnin punaisena lankana 67
KAPPALE 8. Tietokone käsittelee kirjainyhdistelmiä 83
KAPPALE 9. Tietokone ratkaisee matemaattisia ongelmia 101
KAPPALE 10. Tietokone esittää kuvioita 115
KAPPALE 11. Tietokone tallentaa tietoja 129
KAPPALE 12. Tietokone laskee merkkimuuttujilla 139
KAPPALE 13. Tietokone etsii ohjelmointivirheitä 149
KAPPALE 14. Tietokone käsittelee matriiseja 163

LIITE 1. Tietokoneen virheilmoitukset 175
LIITE 2. Tietokoneen vastaukset harjoitustehtäviin 181
LIITE 3. Hakemisto 201

KAPPALE 1

Tietokone otetaan käyttöön

ABC80

ABC 80, joustava tietokonejärjestelmä

ABC 80 on erittäin joustava tietokonejärjestelmä, joka voi koostua monista eri toimintaosista. Yksinkertaisissa sovellustehtävissä, joita käsittelemme kirjan viidessä ensimmäisessä luvussa, käytetään kuitenkin hyväksi vain kahta yksikköä, nimittäin näppäinpöytää ja TV-monitoria. Itse tietokone on piilossa näppäimistön alla.

ABC 80:n perusyksiköt ovat näppäinpöytä ja TV-monitori

Alkuvalmistelut

Ennen kuin johto kytketään sähköverkkoon, on TV-monitori yhdistettävä näppäinpöytään monitorista lähtevällä yhdyskaapelilla.

Yksiköiden liittäminen toisiinsa

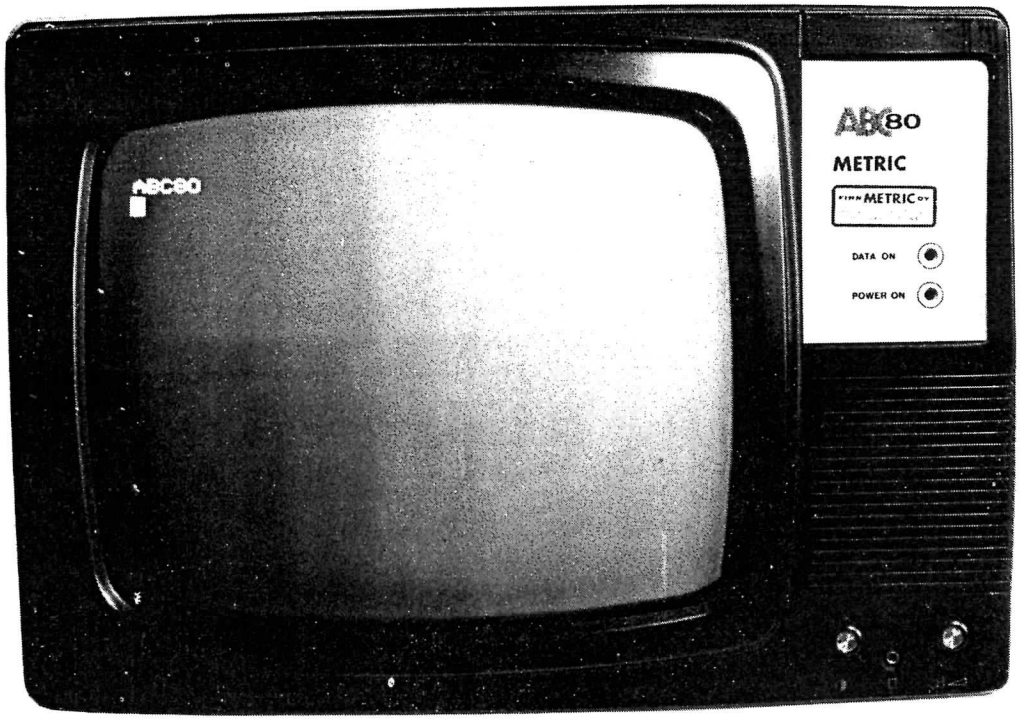
Nyt voit kytkeä TV-monitorin johdon pistorasiaan. Kytke tietokone käyttöön virtakytkimellä, joka sijaitsee TV-monitorin takaosassa. Aseta etuseinässä olevat kaksi säädintä keskiasentoon.

Tietokone on käyttövalmis, kun ABC80 ilmestyy muutaman sekunnin kuluttua kuvaruudun vasempaan ylänurkkaan. Säädä kontrasti siten, että kuva on selkeästi luettavissa.

Tämän jälkeen käytämme seuraavaa kirjoitustapaa ilmaistaksemme kirjoituksen (tekstin) kuvaruudussa:

ABC80

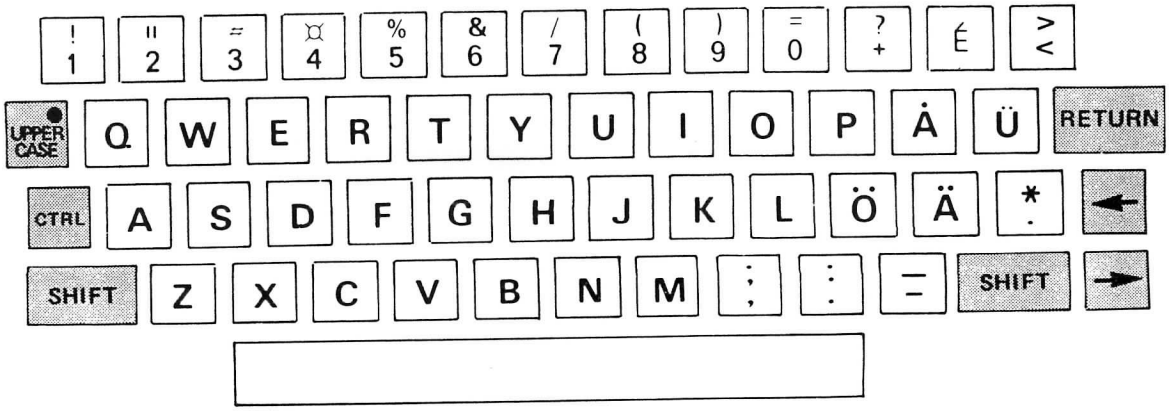
TV-ruudussa näkyvä vilkkuva neliö on nimeltään merkkiosoitin eli osoitin ('kursori', engl. cursor). Osoitin ilmoittaa, mihin kuvaruudun kohtaan tulee se merkki, joka seuraavaksi tulostetaan ruudulle.



ABC 80 valmiina käyttöön

Näppäimistö

ABC 80 -tietokoneen näppäimistö toimii kuin sähkökirjoituskoneen näppäimistö. Kirjoitus näkyy kuvaruudussa, johon mahtuu 24 riviä. Kukin rivi voi sisältää 40 merkkiä. Jos kirjoitat useampia rivejä, vain 24 viimeistä ovat näkyvissä. Kaikki näppäimet ovat toistavia.



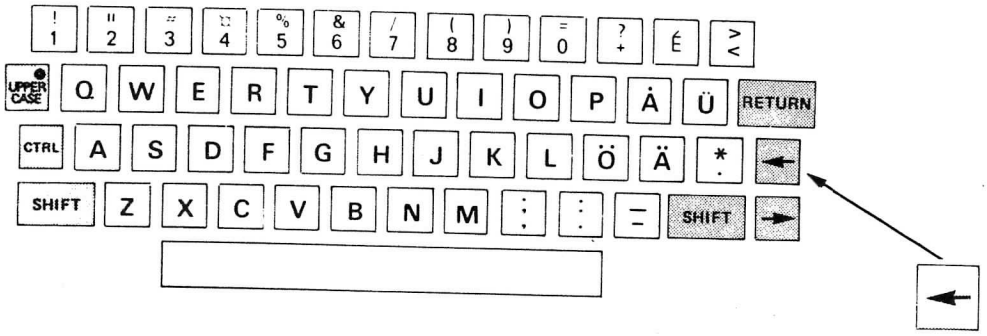
Näppäimistö

Kokeile nyt ABC 80:tä kirjoittamalla näppäimistöllä

```
PRINT
```

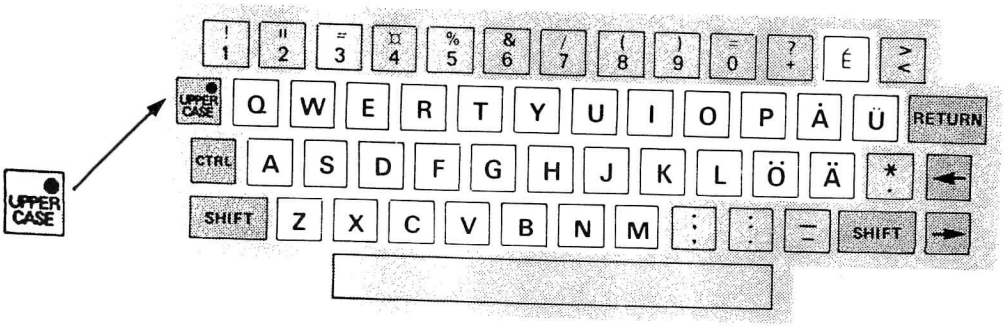
Käytämme jatkossa harmaapohjaista varjostusta ilmaisemaan tekstiä, joka sinun on itse kirjoitettava.

Virhelyönnit voit korjata helposti käyttämällä poistonäppäintä ← (engl. Back Space).



Poistonäppäimen sijainti

Kirjoittamasi virheen voit poistaa näppäilemällä poistonäppäintä sopivan monta kertaa peräkkäin. Aluksi käytämme yksinkertaisuuden vuoksi vain isoja kirjaimia. Ne saat helpoimmin aikaan painamalla näppäintä UPPER CASE . Näppäimessä palava valodiodi ilmaisee, että kone ei kirjoita nyt lainkaan pieniä kirjaimia. Näppäimen UPPER CASE uusi painallus sammuttaa valodiodin ja palauttaa normaalin kirjoituskonetoiminnan sekä pienine että isoine kirjaimineen. Tällöin iso kirjain saadaan aikaan painamalla näppäintä SHIFT samanaikaisesti halutun kirjaimen kanssa.



Näppäimet, joihin UPPER CASE vaikuttaa

Käyttöohjeet

Tämä ABC 80 -tietokoneen käyttökuvaus on tietenkin lyhennelmä, mutta toivottavasti olet nyt kuitenkin oppinut niin paljon, että voit aloittaa koneen käytön. Perusteellisemman teknisen erittelyn löydät koneen mukana tulleesta varsinaisesta käyttöohjeesta.

Muista aina: ennakkoluuloton harjoittelu tuo parhaan tuloksen tietojenkäsittelyyn ja tietokoneen käyttöön perehdyttäessä.

KAPPALE 2

Tietokone älykkäänä kirjoituskoneena

ABC80

Anna tietokoneelle käskyjä

ABC 80 on tietenkin lähes täydellinen tekninen laite, mutta jotta se voisi työskennellä toiveittesi mukaisesti, sinun on selvitettävä sille jollain sopivalla tavalla, mitä haluat sen tekevän. ABC 80 on tehty siten, että se osaa tulkita tiettyjä näppäimistöllä kirjoitettuja käskyjä. Samat käskyt ovat yhteisiä monille mikrotietokoneille. Ne ovat lyhyitä englanninkielisiä ilmauksia.

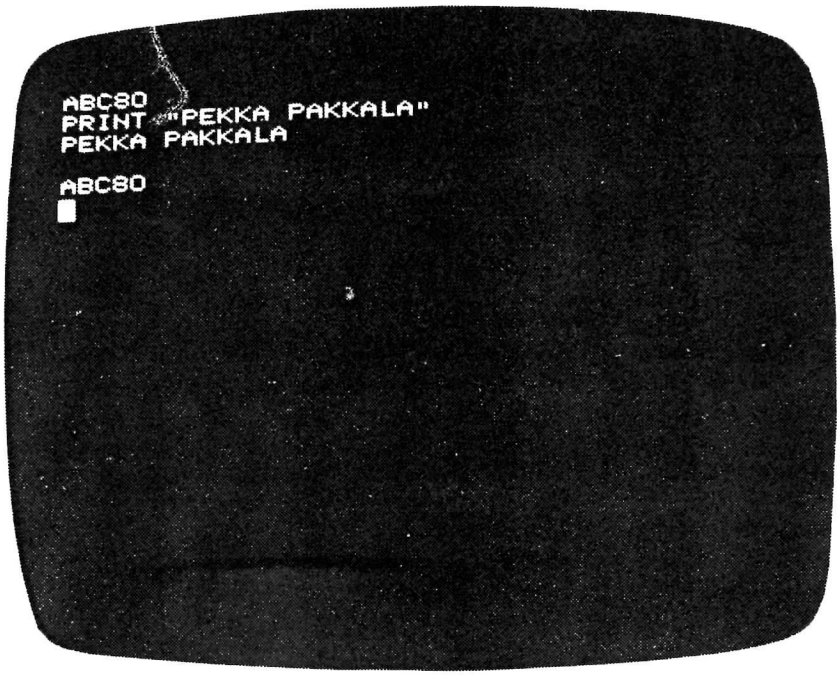
Ensimmäinen yritys

Kuinka konetta on ohjattava, jotta se kirjoittaisi nimesi? Kirjoita silloin aluksi englanninkielinen sana PRINT, joka merkitsee käskyä "kirjoita". Tämän jälkeen näppäile oma nimesi lainausmerkkien ympäröimänä, esimerkiksi

PRINT "PEKKA PAKKALA" [RETURN]

Lainausmerkki saadaan siten, että pidetään näppäintä [SHIFT] alhaalla, samalla kun näppäillään [" 2] .

Kun haluat lopettaa eli 'kuitata' rivin, näppäile *paluunäppäintä* [RETURN] . Tällöin kone tulkitsee ohjeesi ja kirjoittaa antamasi ohjeen mukaisesti nimesi kuvaruutuun.
Kuvaruudussa lukee nyt



TV-ruutu ensimmäisen yrityksen jälkeen

Viimeisellä rivillä oleva ABC80 ilmoittaa, että tietokone on suorittanut antamasi tehtävän ja on valmis ottamaan vastaan uusia toimintaohjeita ja käskyjä. (Useissa muissa tietokoneissa vastaava ilmaisu on READY.)

Pieni ohjelma

Kun haluat kirjoittaa useampia rivejä tietokoneelle tarkoitettuja käskyjä, täytyy jokainen toimintarivi varustaa rivinumerolla. ABC 80 saa tällä tavoin tiedon siitä, missä järjestyksessä käskyt täytetään. Tietokone suorittaa ne yleensä nousevassa numerojärjestyksessä. Tietokoneelle annettua numeroitua käskyrivijonoa sanotaan ohjelmaksi. Kirjoitamme nyt pienen ohjelman siten, että varustamme edellä olleen käskyn rivinumerolla, esimerkiksi 10, ja kirjoitamme sen uudelleen.

```
10 PRINT "PEKKA PAKKALA"
```
RETURN

Jotta et unohtaisi lopettaa jokaista valmiiksikirjoitettua riviä paluunäppäimellä RETURN, merkitsemme sen tässä 2. luvussa jokaisen rivin loppuun. Tietokone ei vielä tiedä, onko ohjelma jo valmis vai onko lisärivejä vielä tulossa. Tämän vuoksi se ei toteuta mitään käskyjä, vaan sinä voit edelleen lisätä uusia rivejä. Voit esimerkiksi käskeä tietokonetta kirjoittamaan myös postiosoitteesi.

```
20 PRINT "02100 ESPOO 10"
```
RETURN

Nyt sinun täytyy selvittää koneelle, että ohjelma on kirjoitettu valmiiksi ja koneen on aika suorittaa se. Tähän käytetään komentoa RUN. Ohjelman suoritusta kutsutaan usein ajoksi.

Kirjoita siis

```
RUN
```
RETURN

TV-kuvaruutuun on nyt ilmestynyt

```
10 PRINT "PEKKA PAKKALA"
20 PRINT "02100 ESPOO 10"
RUN
PEKKA PAKKALA
02100 ESPOO 10

ABC80
```

Ohjelmasi koostuu kahdesta rivistä, joilla on rivinumerot 10 ja 20. Ohjelma säilyy edelleen tietokoneen muistissa, ja voit ajaa sen yhä uudelleen näppäilemällä

```
RUN
```
RETURN

Kokeile tätä muutamia kertoja!

Ohjelma kasvaa

Nyt voit laajentaa ohjelmaasi lisäämällä rivin, joka ilmoittaa lähiosoitteesi.

```
15 PRINT "PÄÄKATU 10" RETURN
```

Aja uusi ohjelma samalla tavalla kuin aikaisemmin.

```
RUN RETURN
```

Mitä tapahtuu? Huomaat, että tietokone toteuttaa käskyt rivinumerojärjestyksessä ja tulos on siten

```
PEKKA PAKKALA
PÄÄKATU 10
02100 ESPOO 10

ABC80
```

Koko ohjelma on edelleen varastossa tietokoneen muistissa. Ohjelman voit tarkistaa komennolla

```
LIST RETURN
```

Kuvaruudussa on nyt ohjelmasi.

```
10 PRINT "PEKKA PAKKALA"
15 PRINT "PÄÄKATU 10"
20 PRINT "02100 ESPOO 10"

ABC80
```

Kuten näet, tietokone on automaattisesti järjestänyt käskyt rivinumeroiden mukaiseen järjestykseen. Ohjelmaa voidaan jatkuvasti laajentaa ja täydentää. Muutakaamme tulostusta siten, että lisätään tyhjä rivi sekä lähiosoitteen edelle että sen jälkeen. Lisää sen vuoksi rivit

```
12 PRINT RETURN
18 PRINT RETURN
```

Aja ohjelma jo opitulla tavalla komennolla

```
RUN RETURN
```

Kuten näet, ovat halutut kaksi tyhjää riviä muodostuneet.

```
PEKKA PAKKALA

PÄÄKATU 10

02100 ESPOO 10

ABC80
```

TV-kuvaruutu tyhjennetään

Voit tyhjentää TV-monitorin tekstistä siten, että pidät paluunäppäintä RETURN painettuna muutaman sekunnin ajan. Tämä ei vaikuta ABC 80 -tietokoneen muistiin, vaan ohjelma säilyy edelleen.

Tarkista asia kirjoittamalla

```
LIST RETURN
```

Ohjelmasi näkyy jälleen kuvaruudussa oikeassa muodossaan.

Rivi poistetaan muistista

Voit poistaa haluamasi ohjelmarivin koneen muistista kirjoittamalla vain sen rivinumeron ja painamalla Return-näppäintä. Kokeile kirjoittamalla 15 RETURN .

Kokeile nytkin ohjelman listausta komennolla LIST.

Muisti tyhjennetään

Jos haluat kirjoittaa kokonaan uuden ohjelman, on sitä ennen kirjoitettava

```
NEW RETURN
```

Käsky NEW tyhjentää tietokoneen muistin. Tämän voit varmistaa kirjoittamalla

```
LIST RETURN
```

Vain **ABC80** ilmestyy kuvaruutuun. Tämä merkitsee sitä, että mitään ohjelmaa ei ole varastoituna tietokoneen muistiin.

Ikuisuuskone?

Kuten olet havainnut, tietokone saadaan yksinkertaisella tavalla kirjoittamaan tekstiä. Lisäksi tietokone järjestää rivit oikeaan järjestykseen automaattisesti.

Seuraava esimerkki osoittaa, että tietokone suoriutuu tästä kaikesta hyvin nopeasti. Aloita uusi ohjelma kirjoittamalla etunimesi näyttöön.

```
10 PRINT "PEKKA"
```
RETURN

Liitä tähän sitten rivi, joka on käskynä tietokoneelle palata takaisin riville 10 ja suorittaa sen käsky uudestaaan.

```
20 GOTO 10
```
RETURN

Ohjelma, jonka olet nyt syöttänyt tietokoneen muistiin, tulee suoritetuksi seuraavalla tavalla:

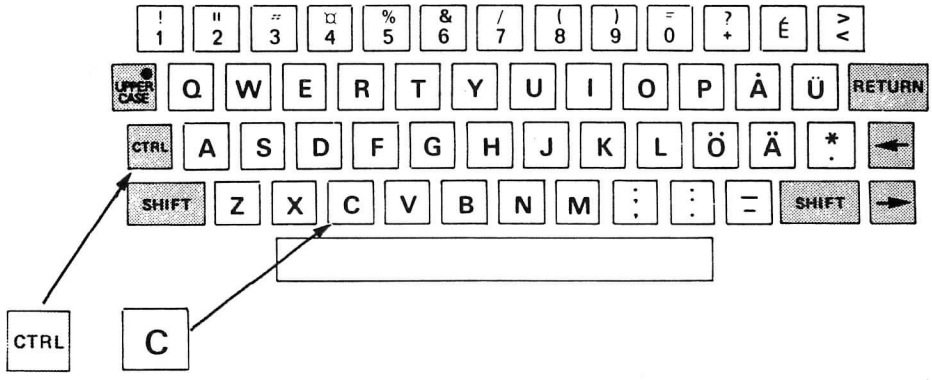
Rivi 10 käskee tietokoneen kirjoittaa nimesi.

Rivi 20 käskee koneen palata takaisin riville 10. Kone kirjoittaa sillä olevan käskyn mukaan nimesi uudestaan, menee sitten riville 20 jne.

Kokeile ohjelmaa kirjoittamalla

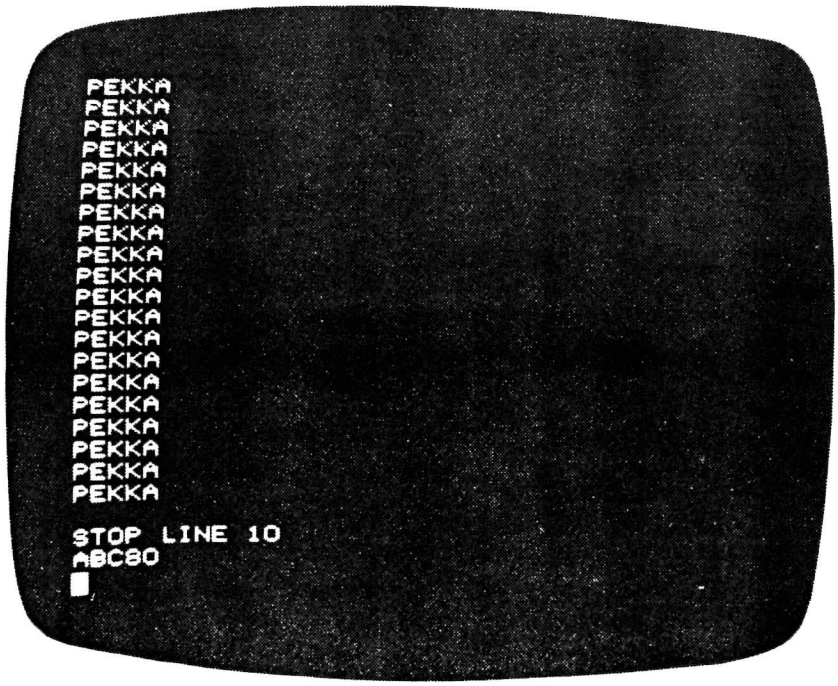
```
RUN
```
RETURN

Kuten huomaat, tietokone täyttää nopeasti TV-ruudun vasemman reunan etunimelläsi ja kirjoittaa sitä taukoamatta. Tietokone on saanut suoritettavakseen tehtävän, jolla ei ole loppua. Voit kuitenkin itse keskeyttää sen painamalla ohjaus- eli kontrollinäppäintä CTRL samanaikaisesti, kun painat näppäintä C.



Ohjelmanajon keskeyttäminen

ABC 80 kirjoittaa tällöin näkyviin sen ohjelmarivin numeron, jossa sen toiminta keskeytettiin. Samanaikaisesti se antaa äänisignaalin kaiuttimesta.



Keskeytetty tulostus

Näytössä oleva STOP LINE 10 ilmoittaa, että tietokone keskeytettiin, kun se oli suorittamassa riviä 10.

Tulostus samalle riville

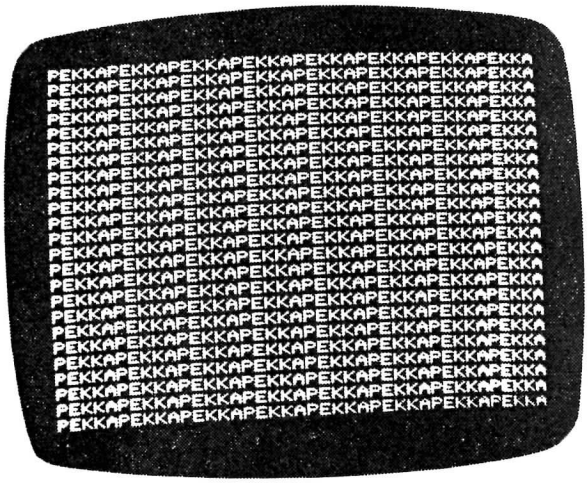
Jos et halua tietokoneen aloittavan uutta riviä joka kerta, kun se on suorittanut rivillä 10 annetun tehtävän edellisen ohjelman mukaisesti, voit helposti muuntaa ohjelman tulostusmuotoa.

Kirjoita rivi 10 uudestaaan, mutta lisää puolipiste rivin loppuun.

```
10 PRINT "PEKKA";
```
RETURN

Huomaa, mitä tapahtuu, kun kirjoitat

```
RUN
```
RETURN



Puolipiste antaa yhtäjaksoisen eli "pakatun" tulostuksen.

Tulostus syntyy niin nopeasti, että sitä on joskus vaikea käsittää. Keskeytä ohjelmanajo kuten ennenkin näppäilemällä CTRL ja C . Jos päätät PRINT-käskyn puolipisteellä, niin normaali rivinsiirto poistuu, ja teksti tulee kirjoitetuksi peräkanaa ilman välejä.

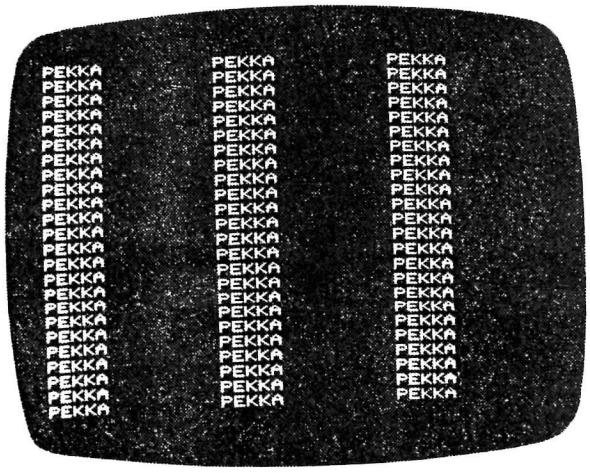
Tulostus taulukko- eli sarakemuodossa

Kirjoita jälleen rivi 10 uudestaan, mutta vaihda lopussa oleva puolipiste pilkuksi.

```
10 PRINT "PEKKA", RETURN
```

Aja ohjelma tavanomaiseen tapaan.

```
RUN RETURN
```



Pilkku tulostaa tiedot sarakkeisiin

Keskeytä se näppäilemällä CTRL ja C .

Kuten näet kuvaruudusta, teksti tulostetaan nyt sarakkeisiin. Jos lopetat PRINT-rivin pilkulla, niin normaali rivikirjoitus muutetaan muotoon, jossa on peräkkäin kolme toisiaan seuraavaa saraketta. Kuvaruutu jakautuu siis kolmeksi sarakkeeksi. Näin yksinkertaisesti voidaan tehdä pieniä taulukoita.

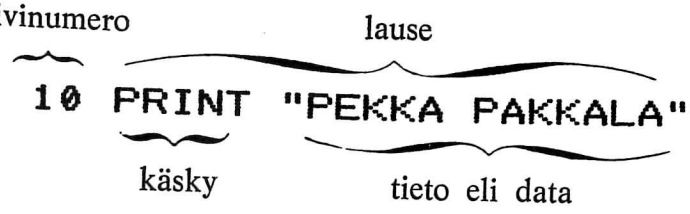
Tietokonekieli BASIC

Tietokonekieli on tietty järjestelmä sääntöjä ja ilmauksia; ne ovat toimintakäskyjä ja komentoja, joita tietokone ymmärtää. Se, kuten kaikki muutkin kielet, sisältää määrätyn sanavaraston ja niiden käyttöä säätelevän kieliopin. Tietokonekieltä, jota me käytämme ABC 80 -tietokoneen ohjelmoimiseksi, kutsutaan nimellä BASIC (Beginners All-Purpose Symbolic Instruction Code). BASIC-kieli kehitettiin 1960-luvulla Darthmoutin Collegessa. Tavoitteena oli, että kuka tahansa voisi sen avulla käyttää tietokoneita. Muihin tietokonekieliin verrattuna on BASIC-kielellä monia etuja. Se on helppo oppia, ohjelmointi tapahtuu "keskustelemalla" koneen kanssa ja lisäksi on saatavissa lukuisia valmiita ohjelmia. Itse asiassa sinäkin olet tähän mennessä käyttänyt paria valmista ohjelmaa. Esimerkiksi

```
10 PRINT "PEKKA PAKKALA"
```

on yhden rivin BASIC-kielinen tietokoneohjelma.

Helpoimmin BASIC-ohjelma tunnistetaan siitä, että käskyjä käytetään muodostamaan lauseita, jotka kirjoitetaan riveittäin rivinumeroilla varustaen.



Komennot

Itse BASIC-ohjelmaan liittyvien käskyjen lisäksi täytyy tietokoneelle antaa komentoja. Niillä ilmoitetaan, kuinka tietokoneen on käsiteltävä itse ohjelmaa. Tähän mennessä olemme käyttäneet esimerkiksi komentoja RUN, LIST ja NEW.

Suurta osaa komennoista ei voi käyttää käskyinä itse ohjelmassa.

	Selitys
`10 PRINT "PEKKA PAKKALA"`	Ohjelma
`RUN`	Komento
`PEKKA PAKKALA`	Tulos
`ABC80`	Kuittaus; ohjelma on suoritettu loppuun ja kone on valmis ottamaan vastaan uusia lauseita tai komentoja.

Yhteenveto

Tästedes päätämme jokaisen luvun yhteenvedolla uusista BASIC-käskyistä samoin kuin luvussa esitellvistä komennoista

Käsky	Selitys	Esimerkki
`PRINT`	Tulostaa kuvaruutuun. Teksti on ympäröitävä lainausmerkeillä.	`10 PRINT "HEI"`
	Tekstin jäljessä oleva puolipiste eliminoi eli poistaa rivinsiirto-toiminnan.	`10 PRINT "HEI";`
	Tekstin jäljessä oleva pilkku määrää seuraavan tekstin seuraavaan sarakkeeseen.	`10 PRINT "HEI",`
`GOTO`	Määrää ohjelman jatkumaan tietyltä riviltä.	`20 GOTO 10`

Komento	Selitys	Esimerkki
`RUN`	Ajaa l. suorittaa muistissa olevan ohjelman.	`RUN`
`NEW`	Poistaa (tyhjentää) ohjelman muistista.	`NEW`
`LIST`	Antaa näyttöön muistissa olevan ohjelman. Eri vaihtoehtoja voidaan käyttää:	
	Kaikki rivit	`LIST`
	Vain yksi rivi	`LIST 100`
	Kaikki kahden rivinumeron	`LIST 100-200`

väliltä (nämä mukaan-
luettuina)

Kaikki alusta tiettyyn riviin saakka (tämä mukaanluettuna) `LIST -200`

Kaikki tietystä rivistä lähtien `LIST 100-`

CTRL C Keskeyttää ohjelman ajon

Harjoituksia

Huomaa: Muista käyttää komentoa NEW aina ennen uutta harjoitustehtävää!

2.1 Mitä tulostaa seuraava ohjelma?

```
10 PRINT "MITÄ"
20 PRINT "TÄMÄ"
30 PRINT "TULOSTAA ?"
```

2.2 Millaisen tulostuksen antaa puolestaan seuraava ohjelma?

```
10 PRINT "ENTÄ",
20 PRINT "SITTEN",
30 PRINT "TÄMÄ ?"
```

2.3 Kuinka on harjoituksen 2.1 ohjelmaa muutettava, jotta näyttöön tulisi seuraava ilmaus?

```
MITEN TÄMÄ TULOSTETAAN ?
```

Huomaa, että sanojen välissä on välilyönti! (SPACE)

2.4 ABC 80 -tietokoneen näppäimistöllä kirjoitetaan seuraavat rivit:

```
10 PRINT "PEKKA PAKKALA"
20 PRINT "02100 ESPOO 10"
15 PRINT "PÄÄKATU 10"
10 PRINT "ANNA PAKKALA"
```

Mitä tapahtuu, kun ohjelma ajetaan käskyllä RUN?

2.5 Mihin kohtaan kuvaruutua saadaan näyttö seuraavalla ohjelmalla?

```
10 PRINT,"LIISA PAKKALA"
20 PRINT,"PÄÄKATU 10"
30 PRINT,"02100 ESPOO 10"
```

2.6 Täydennä seuraavassa ohjelmassa käskyjen GOTO perästä puuttuvat rivinumerot, kun näyttöön halutaan tulostus OK!

```
10 GOTO ......
20 PRINT "K" ;
30 GOTO ......
40 PRINT "O" ;
50 GOTO ......
60 PRINT "!"
```

Tarkista kokeilemalla.

2.7 Mihin sarakkeeseen tulostuu puhelinnumero seuraavassa ohjelmaesimerkissä?

```
10 PRINT "PEKAN PUH.N:O ON ",
20 PRINT "123456"
```

KAPPALE 3

Tietokone etsii tietoja

ABC80

Nimestä tulee merkkijonomuuttuja

Jos käytät ohjelmassa usein esimerkiksi nimeä, voit sen sijasta käyttää lyhyempää merkintää, jota kutsutaan muuttujaksi. Muuttujaa, joka voi sisältää sekä kirjaimia että numeroita, kutsutaan merkkijonomuuttujaksi ja se merkitään yhdellä kirjaimella ja 'valuuttamerkillä' ¤ (luetaan esim. aurinko tai eurovaluutta) esimerkiksi A¤. ¤ on uusi pohjoismainen normi, jota useissa muissa laitteissa vastaa merkki $ eli dollari. Siitä nimitys 'valuuttamerkki'. Merkin ¤ kirjoitat käyttämällä näppäimiä [SHIFT] ja [¤ 4]. Jos haluat usein käyttää omaa nimeäsi ohjelmassa, voit antaa nimesi merkkijonomuuttujan arvoksi. Tämä arvon sijoittaminen tapahtuu BASIC-käskyllä LET, joka merkitsee "sijoita". Ennen kuin tarkastelemme, kuinka käskyä LET käytetään, täytyy tietokoneen muisti tyhjentää.

Kirjoita siksi

```
NEW
```

(Tämän jälkeen emme enää muistuta paluukomennosta RETURN, joka vaaditaan jokaisen rivin lopussa.)

Seuraava ohjelmarivi antaa muuttujalle A¤ arvon PEKKA PAKKALA:

```
10 LET A¤="PEKKA PAKKALA"
```

Huomaa, että nimen ympärille on kirjoitettava lainausmerkit. Osoitamme nyt, että muuttujasymboli A¤ merkitsee rivin 10 suorittamisen jälkeen nimeä PEKKA PAKKALA.

Lisää ohjelmaan rivi

```
20 PRINT A¤
```

Aja ohjelma kirjoittamalla

```
RUN
```

Ajon jälkeen saat tulostukseksi nimen. Kuvaruudussa on esimerkiksi

```
10 LET A¤=PEKKA PAKKALA"
20 PRINT A¤
RUN
PEKKA PAKKALA

ABC80
```

Tiedonsyöttö ohjelman suorituksen aikana

Merkkijonomuuttujalle A¤ voidaan sijoittaa arvo muillakin tavoin kuin käskyllä LET. Tarkastelkaamme erästä vaihtoehtoa. Muuta rivi 10 kirjoittamalla tilalle uusi rivi samalla rivinumerolla 10.

```
10 INPUT A¤
```

Kun tietokone tulee ohjelmanajossa riville 10, ajo keskeytyy, kunnes syötät näppäimistöllä merkkijonomuuttujan A¤ arvon. Tarkista nyt, että olet kirjoittanut ohjelman virheettömästi.

```
LIST
10 INPUT A¤
20 PRINT A¤

ABC80
```

Aja ohjelma komennolla RUN tavalliseen tapaan. Mitä tapahtuu? Huomaat, että tietokone kirjoittaa eli tulostaa kysymysmerkin kuvaruutuun. Se merkitsee sitä, että sinun on nyt syötettävä muuttujalle A¤ jokin arvo näppäimistöllä. Kirjoita esimerkiksi oma nimesi ja muista, että se on kirjoitettava lainausmerkkien sisään, mikäli haluat myös mahdollisten välilyöntien tulevan esiin etu- ja sukunimien välillä.

Kun nyt näppäilet [RETURN] nimesi syötön jälkeen, niin tietokone jatkaa ohjelmansuoritusta riviltä 20. Se tulostaa merkkijonomuuttujan A¤ arvon eli sinun nimesi kuvaruutuun.

Useita muuttujia samassa lauseessa

Jos haluat samanaikaisesti syöttää eli sijoittaa useita merkkijonomuuttujan arvoja yhdellä INPUT-lauseella, riittää, että eri muuttujat erotetaan toisistaan pilkulla.

```
10 INPUT A¤,B¤
20 PRINT A¤,B¤
```

Kun nyt ajat ohjelman, niin kuvaruutuun ilmestyy kysymysmerkki kuten ennenkin. Syötä tällöin muuttujien arvot A¤ ja B¤ , esimerkiksi nimesi ja puhelinnumerosi pilkulla toisistaan erotettuina.

```
?"PEKKA","123456"
```

Käskyllä RETURN tietokone kirjoittaa muuttujien A¤ ja B¤ arvot ohjelma-

rivin 20 mukaisesti. Koska muuttujien A¤ ja B¤ välissä oli pilkku, niin ensin mainitun arvo kirjoitetaan sarakkeeseen 1 ja jälkimmäisen arvo sarakkeeseen 2.

```
PEKKA          123456

ABC80
```

Sijoitus Read-käskyllä

Jos haluat sijoittaa ohjelmassa eri arvoja merkkijonomuuttujille katkaisematta ohjelmanajoa, voit käyttää käskyjä READ ja DATA.

Tarkastelkaamme esimerkkiä.

```
10 READ A¤,B¤
20 DATA "PEKKA","123456"
30 PRINT A¤,B¤
```

Kun määräät ajon komennolla RUN, tapahtuu seuraavaa. Rivi 10 suoritetaan, ja tietokone etsii ensimmäisen vastaantulevan DATA-lauseen ohjelmasta. Kun se löytää sellaisen, DATA-lauseen ensimmäinen arvo (PEKKA) sijoitetaan muuttujan A¤ arvoksi. DATA-lauseen toinen arvo (123456) sijoitetaan vastaavasti muuttujan B¤ arvoksi. Rivillä 30 olevalla PRINT-lauseella tulostetaan sijoitetut arvot näyttöön. READ- ja DATA-käskyt toimivat aina yhdessä, eivät koskaan erikseen.

Tietokone ilmoittaa tekemäsi virheen

Mitä tapahtuu, jos lisäät seuraavan rivin?

```
40 GOTO 10
```

Tietokone menee takaisin ohjelmariville 10 ja yrittää lukea uudet arvot muuttujille A¤ ja B¤ . Mutta rivin 20 DATA-lause tuli luetuksi loppuun jo ensimmäisellä kerralla eikä uutta DATA-lausetta tietoineen ole koko ohjelmassa.

Kokeile tätä kirjoittamalla RUN. Tällöin ABC 80 reagoi äänisignaalilla ja tulostamalla

```
PEKKA          123456

ERR 30 LINE 10
ABC80
```

Virheilmoitus annetaan aina, kun jokin epälooginen tai kielletty toiminto käsketään suoritettavaksi. Katso eri virheilmoitukset tarkemmin liitteestä 1.

Varmistaaksemme sen, että uusien DATA-lauseiden tiedot todellakin sijoitetaan edelleen muuttujille A¤ ja B¤ , lisäämme ohjelmaan rivit

```
21 DATA "OLLI","234567"
29 DATA "KAARINA","345678"
```

Kirjoita LIST ja varmistaudu, että ohjelma näyttää seuraavalta:

```
LIST
10 READ A¤,B¤
20 DATA "PEKKA","123456"
21 DATA "OLLI","234567"
29 DATA "KAARINA","345678"
30 PRINT A¤,B¤
40 GOTO 10

ABC80
```

Kun ajat ohjelman, saat tulokseksi

```
PEKKA          123456
OLLI           234567
KAARINA        345678

ERR 30 LINE 10
ABC80
```

Kuten huomaat ja kuulet, annetaan nytkin virheilmoitus. Se kertoo tietokoneen yrittävän lukea muuttujille A¤ ja B¤ enemmän arvoja kuin olemme antaneet DATA-lauseissa. Tämä johtuu siitä, että tietokone siirtyy edelleen riviltä 40 riville 10 myös silloin, kun kaikki tiedot on käytetty. Virhe voidaan korjata lisäämällä rivit

```
35 IF A¤="KAARINA" THEN GOTO 100
100 PRINT "SELVÄ"
```

Lisärivi 35 sisältää sen, että ABC 80 testaa, onko muuttujan A¤ arvoksi jo sijoitettu KAARINA viimeisimmästä DATA-lauseesta. Jos A¤ :n arvo ei ole KAARINA, niin tietokone jatkaa seuraavalta riviltä ja siirtyy sen käskyn mukaisesti riville 10. Jos sitävastoin A¤ :lle on sijoitettu arvo KAARINA, *niin* tietokone siirtyykin riville 100. Se on PRINT-lause, joka tulostaa näyttöön SELVÄ. Rivin 100 jälkeen ei ole lisärivejä, joten ohjelma on loppuun suoritettu. Aja ohjelma *ja totea, ettet* nyt saa virheilmoitusta.

```
RUN
PEKKA          123456
OLLI           234567
KAARINA        345678
SELVÄ

ABC80
```

Voidaksemme kerrata, mitä ohjelmassa tapahtuu, kirjoita LIST.

```
LIST
10 READ A¤,B¤
20 DATA "PEKKA","123456"
21 DATA "OLLI","234567"
29 DATA "KAARINA","345678"
30 PRINT A¤,B¤
35 IF A¤="KAARINA" THEN GOTO 100
40 GOTO 10
100 PRINT "SELVÄ"

ABC80
```

Rivillä 10 sijoitetaan muuttujille A¤ ja B¤ arvot ensimmäisestä DATA-lauseesta käskyllä READ. Lauseet READ ja DATA esiintyvät aina yhdessä. DATA-lauseet voidaan sijoittaa mihin tahansa ohjelmassa ja niitä voi olla mielivaltaisen monta. Käskyllä READ haetaan arvot DATA-lauseista nousevan rivinumeron mukaan eli kone muistaa, miltä riviltä viimeksi luettiin. Rivi 30 tuottaa näyttöön muuttujien A¤ ja B¤ senhetkiset arvot. Rivillä 35 tietokone tutkii, onko muuttujalle A¤ sijoitettu arvo viimeisestä DATA-lauseesta. Jos niin on, tietokone siirtyy riville 100 ja kirjoittaa SELVÄ. Muussa tapauksessa se jatkaa riville 40, jolta siirrytään riville 10 hakemaan uusia muuttujanarvoja.

Pieni puhelinluettelo

Kirjoittamamme ohjelma on itse asiassa ohjelma, joka kirjoittaa näyttöön pienen puhelinluettelon. Voit helposti laajentaa luetteloa lisäämällä uusia DATA-lauseita, joissa on kaikissa nimi ja puhelinnumero. Esimerkiksi.

```
22 DATA "ERKKI","122133"
23 DATA "KAUKO","433415"
24 DATA "LEENA","545325"
```

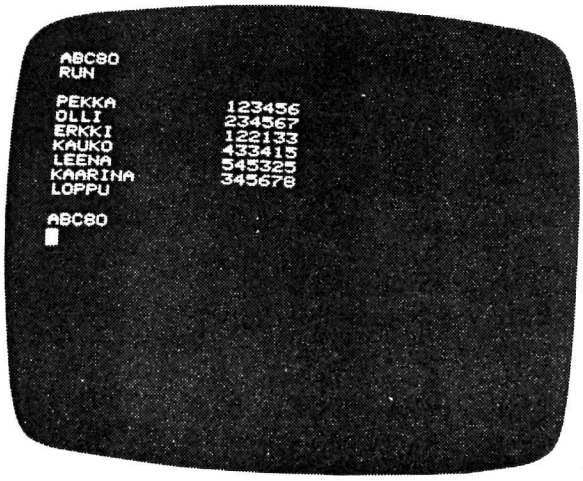
Jos kirjoitat DATA-lauseita, joilla on suurempi rivinumero kuin 29, on sinun muutettava myös riviä 35 siten, että testi kohdistuu viimeiseen arvoon, jonka muuttuja A¤ voi saada.

Ilmavampi näyttö

Tehdäksemme TV-ruudussa olevan näytön ilmavammaksi voimme tehdä tyhjän rivin tulostuksen alkuun kirjoittamalla

```
5 PRINT
```

Aja ohjelma käskyllä RUN ja katso omaa puhelinluetteloasi.



Puhelinluettelon tiedot tulostettuina

Tietokone etsii puhelinluettelosta

Kehittäkäämme ohjelmaamme edelleen niin, että tietokone itse etsii meille halutut puhelinnumerot luettelostaan.

Lisää rivit

```
6 PRINT "KENEN PUHELINNUMERON HALUAT ?"
7 INPUT N¤
```

Rivi 6 sisältää näyttöön tulevan selventävän tekstin ennen seuraavana olevaa INPUT-lausetta.

Rivi 7 tulostaa näyttöön kysymysmerkin, jonka jälkeen tietokone odottaa, että muuttujalle N¤ annetaan arvo näppäimistöllä. Muuttujan N¤ arvoksi sijoitetaan sen henkilön nimi, jonka puhelinnumero halutaan tietää.

Korvaa vanha rivi 30 uudella kirjoittamalla

```
30 IF N¤=A¤ THEN GOTO 50
```

Rivillä 30 tietokone testaa, onko syötetty nimi sama kuin se, joka on sillä hetkellä muuttujan A¤ arvona. Jos N¤ = A¤ , niin nimeen liittyvä puhelinnumero B¤ kirjoittuu TV-ruutuun. Se on etsitty puhelinnumero. Rivin 30 mukaisesti tietokone siirtyy riville 50, jos se löysi haetun henkilön puhelinnumeron. Rivin 50 täytyy siis sisältää PRINT-käsky, joka määrää kirjoittamaan näyttöön oikean puhelinnumeron.

```
50 PRINT "NUMERO ON",B¤
```

Jos sen sijaan rivillä 30 ei ole voimassa N¤ = A¤ , niin tietokone jatkaa riville 35. Siellä testataan, onko muuttujalle A¤ sijoitettu jo puhelinluettelon viimeinen nimi. Jos niin ei ole, niin tietokone jatkaa rivin 40 kautta riville 10. Rivillä 10 muuttujille sijoitetaan uudet arvot seuraavasta DATA-lauseesta.

Valmis ohjelmasi näyttää nyt seuraavalta, mahdollisesti DATA-lauseita lukuunottamatta, joiksi sinulla lienee oma ehdotus.

```
LIST

5 PRINT
6 PRINT "KENEN PUHELINNUMERON HALUAT ?"
7 INPUT N¤
10 READ A¤,B¤
20 DATA "PEKKA","123456"
21 DATA "OLLI","234567"
22 DATA "ERKKI","122113"
23 DATA "KAUKO","433415"
24 DATA "LEENA","545325"
29 DATA "KAARINA","345678"
30 IF N¤=A¤ THEN GOTO 50
35 IF A¤="KAARINA" THEN GOTO 100
40 GOTO 10
50 PRINT "NUMERO ON",B¤
100 PRINT "SELVÄ"

ABC80
```

Esimerkki ohjelman ajosta ylläolevin DATA-lausein:

```
RUN
KENEN PUHELINNUMERON HALUAT ?
? LEENA
NUMERO ON       545325
SELVÄ

ABC80
```

Aja ohjelma useita kertoja ja kokeile eri nimillä, jotka löytyvät DATA-lauseista.

Mitä tapahtuu, jos vastaat nimellä, jota ei ole DATA-lauseissa? Silloin tietokone kirjoittaa SELVÄ, koska tietokone siirtyy riville 100, kun rivin 35 ehto tulee täytetyksi. Näin tapahtuu, kun kaikki DATA-lauseet on kokeiltu. Ohjelman täydentäminen on harjoitustehtävänä 3.1.

Tietorekisteristä hakeminen, tyypillinen atk-tehtävä

Ohjelman, joka hakee oikean puhelinnumeron luettelosta, ei ilmeisesti tarvitse olla erityisen monipuolinen, ja sitä voidaan helposti laajentaa mielivaltaisen moneen DATA-lauseeseen. Normaalisti, kun tietokoneet hakevat tietoja suuremmista rekistereistä, ei tietoja tallenneta DATA-lauseina vaan muilla tavoin. Tiedon haun periaate on kuitenkin sama, vaikkakin tiedon tallennusmuodot voivat vaihdella. Olethan varmasti nähnyt joko omalla tai perheenjäsenesi nimellä varustettuja postilähetyksiä, joihin nimet on poimittu tiettyjen ehtojen — esim. postinumeron mukaan — jostakin suuresta tietorekisteristä. Tässä luvussa olevat esimerkit kuvaavat sen vuoksi hyvin tavanomaista ja jatkuvasti kasvavaa tietokoneiden käyttöaluetta.

Yhteenveto

Käsky	Selitys	Esimerkki
LET	Sijoittaa muuttujalle arvon. ABC 80 BASIC sallii myös LET-sanan poisjättämisen.	`10 LET A$="KALLE"` `10 A$="KALLE"`
INPUT	Odottaa tiedon syöttämistä näppäimistöllä. Tieto on syötettäessä sijoitettava sitaatteihin vain, jos jonossa on välilyöntejä, pilkkuja tai lainausmerkkejä.	`10 INPUT A$`
READ	Lukee muuttujanarvon DATA-lauseesta.	`10 READ A$`
DATA	Tallentaa muuttujan arvon, joka luetaan READ-lauseella. Kun merkkijonossa ei	`20 DATA "HEIPÄ HEI"`

ole välilyöntejä, pilkkuja eikä lainausmerkkejä, ei jonoa tarvitse sijoittaa lainausmerkkeihin. DATA-lauseet saavat olla ohjelman missä tahansa kohdassa.

```
30 DATA KALLE
```

IF...THEN... Jos ehto on täytetty, ohjelma jatkuu käskyllä, joka on ilmoitettu sanan THEN jäljessä. Muutoin jatketaan seuraavalla rivillä.

```
20 IF A$=B$
THEN GOTO 100
```

Ylläolevia BASIC-käskyjä voidaan käyttää myös käsiteltäessä muuttujia, joiden arvoina on pelkkiä lukuja. Esimerkkejä tästä on 4 luvussa.

Harjoituksia

3.1 Sivulla 26 oleva ohjelma tulostaa tekstin SELVÄ myös silloin, kun kysymme sellaisen henkilön puhelinnumeroa, jota ei ole rekisterissä. Täydennä ohjelmaa siten, että tällaisessa tapauksessa saadaan näyttöön tulostus EI LÖYDY LUETTELOSTAMME.

3.2 Muuta tehtävän 3.1 ohjelmaa siten, että etsiminen tapahtuu nimen sijasta numeron perusteella. Ohjelma tulostakoon sen henkilön nimen, jolla on syötetty puhelinnumero.

3.3 Kaukonäköinen käytettyjen autojen kauppias, joka on hankkinut tietokoneen, on kirjoittanut allaolevan ohjelman helpottaakseen asiakkaan valintaa suuresta valikoimasta.

```
10 PRINT
20 PRINT "HALUTTU AUTOMERKKI ?"
30 INPUT A$
40 PRINT
50 PRINT "MERKKI","VÄRI","AJETTU (km)"
60 PRINT
70 READ B$,C$,D$
80 IF A$=B$ THEN GOTO 200
90 IF B$="LOPPU" THEN GOTO 220
```

```
100 GOTO 70
110 DATA "FIAT","SININEN","40000"
120 DATA "SAAB","VIHREÄ","15000"
130 DATA "VOLVO","VIHREÄ","10000"
140 DATA "SAAB","KELTAINEN","20000"
150 DATA "VOLVO","PUNAINEN","120000"
190 DATA "LOPPU","LOPPU","LOPPU"
200 PRINT B$,C$,D$
210 GOTO 70
220 PRINT
```

Ohjelman tarkoitus on, että asiakas saisi listan, joka sisältää kaikki tietynmerkkiset autot, jotka ovat varastossa. Erään ajon tulos näyttää seuraavalta:

```
HALUTTU AUTOMERKKI ?
? VOLVO

MERKKI          VÄRI            AJETTU (km)

VOLVO           VIHREÄ          10000
VOLVO           PUNAINEN        120000
```

Eräänä päivänä autokauppias hankkii seuraavat kaksi autoa: punainen Volvo, jolla on ajettu 30 000 km ja sininen Toyota, jolla on ajettu 60 000 km. Kirjoita ohjelmaan kaksi DATA-lausetta sopivaan paikkaan niin, että rekisteri on ajan tasalla, ja aja ohjelma uudestaan.

Kuinka ohjelmaa on muutettava, kun autokauppias myy vihreän Volvonsa, jolla on ajettu 10 000 km?

3.4 Edellä esitelty autokauppias huomaa vähitellen, että monet asiakkaat ovat kiinnostuneempia ajattelemansa auton väristä kuin mallista. Kuinka ohjelmaa on muunnettava, jotta se tulostaisi listan kaikista tietynvärisistä autoista?

KAPPALE 4

Tietokone suorittaa laskuja

ABC80

Tietokone vai laskin?

Tähän mennessä et ole käyttänyt tietokonetta mihinkään laskutehtäviin, mutta luonnollisesti se on myös käyttökelpoinen, nopea ja pitkälle kehitetty laskin. Tietokoneen suuri etu laskimeen on kuitenkin se, että tulos voidaan esittää paljon havainnollisemmin, koska tietokone voi esittää kuvaruudussa sekä laajalti tekstiä että useita lukuja samanaikaisesti. Laskinhan näyttää tavallisesti vain yhden ainoan luvun kerrallaan.

Tietokoneen tekemä tiliote

Tiliote, jonka saat pankista, ilmoittaa esimerkiksi šekkitilisi tilitapahtumat. Tiliotteen numerotiedot ovat yleensä tietokoneen laskutoimitusten tuloksia.

Pvm. Datum	Selite Förklaring	Arvop. Val.dag	Otot Uttag	Panot Insättningar
	SALDO 22.01.81			112927 96
2301	MAKSUPALVELU1040	2301	216 00	
2301	OMA PANO 4121	2601		16200 00
2301	SHEKKI 698 9992	2201	18000 00	
	TAPAHT. YHT.		18216 00	16200 00
	VUODEN ALUSTA		167856 21	275685 80
	SALDO 23.01.81			110911 96

Esimerkki tiliotteesta

Yksinkertainen BASIC-ohjelma, jota voit itse käyttää pankkitilillä tapahtuvien muutosten kirjaamiseen, voi näyttää seuraavalta:

```
10 PRINT
20 PRINT "ANNA NYKYINEN SALDO";
30 INPUT A
40 PRINT "ANNA TAPAHTUMA";
50 INPUT B
60 PRINT
70 LET A=A+B
80 PRINT "TAPAHTUMA","UUSI SALDO"
90 PRINT B,A
100 PRINT
110 GOTO 40
```

Nyt on aika ottaa esille tiliote ja kokeilla ohjelmaa. Esimerkiksi 80 mk:n talletus syötetään luvulla 80. Esimerkiksi 100 mk:n nosto syötetään luvulla —100. Varat ja velat kuvataan vastaavalla tavalla saldossa.

Selvitä käyttöohjeesta, missä on näppäin —.

Esimerkki:

Alkusaldo	1000
Tilitapahtuma	—1500
Uusi saldo	— 500

Edellä esitetyn ohjelman selitys on seuraava:

Rivinumero	**Rivin toiminta**
10, 60, 100	Tulostetaan tyhjä rivi asioiden erottamiseksi toisistaan.
20, 40	Tulostaa johdantotekstin, joka kertoo, mitkä tiedot kulloinkin tulee antaa.
30, 50	Tietylle muuttujalle syötetään arvo näppäimistöllä.
70	LET A = A + B merkitsee sitä, että vanha saldo A korvataan vanhan saldon A ja tilitapahtuman B summalla. Vanha saldo (A:n aikaisempi arvo) näkyy edelleen kuvaruudussa.
80	Teksti joka selventää tulostuksen.
90	Tulostaa muuttujien B ja A arvot. TV-kuvaruutuun kirjoitetaan siis tilitapahtumaa ja saldoa vastaavat rahasummat.
110	Kun riveillä 10—100 annetut tehtävät on suoritettu, tietokone palaa suorittamaan rivin 40 käskyä.

Tietokone suorittaa ohjelman käskyjä niin kauan kuin jatkat uusien tilitapahtumien syöttämistä. Voit keskeyttää ohjelmanajon jättämällä syöttämättä kysytyn luvun ja painamalla pelkästään paluunäppäintä (RETURN). Tietokone tulkitsee tämän virhetoiminnaksi. Se keskeyttää ohjelmanajon ja antaa virheilmoituksen

```
ERR 12 LINE 50
ABC80
```

Kuten voit liitteestä 1 tarkistaa, merkitsee ERR 12 "Virheellinen luku", mikä tässä tapauksessa tarkoittaa luvun puuttumista kokonaan.

Laskutoimituksissa käytetään lukumuuttujaa

Kuten näit, edellisessä ohjelmassa muuttujista puuttui ¤-merkki. Kutsumme merkkiä ¤ jonomerkiksi ja sisältävää muuttujaa merkkijonomuuttujaksi. Jonomuuttujassa voidaan käyttää sekä numeroita, kirjaimia että muita merkkejä. Muuttujaa, jossa puolestaan ei käytetä jonomerkkiä, sanotaan lukumuuttujaksi. Se voi edustaa vain lukua. Laskutoimituksissa käytetään yleensä vain lukumuuttujia. Tapauksia, joissa laskutoimituksissa käytämme merkkejä lukujen asemasta, tarkastelemme myöhemmin.

Vaivaton tapa korjata ja muuttaa ohjelmia

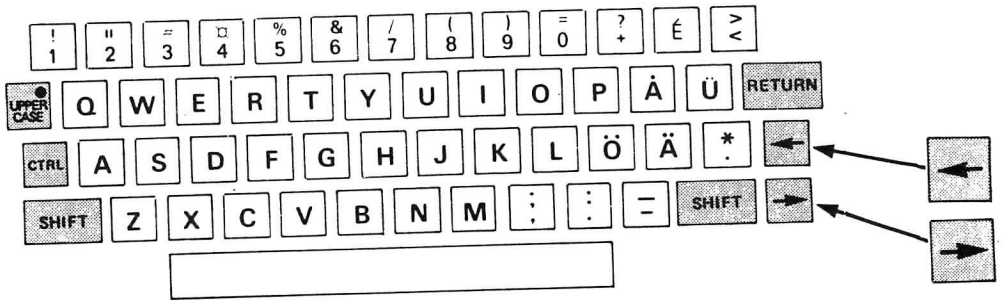
Luvussa 1 olet oppinut, kuinka poistonäppäintä käytetään virhelyönnin korjaamiseen. Kuinka muutos tehdään jo valmiiksikirjoitettuun ohjelmaan? Eräs mahdollisuus on kirjoittaa koko rivi uudestaan muutetussa muodossa. Komento ED (lyhenne englanninkielen sanasta EDIT) tarjoaa kuitenkin paljon mukavamman mahdollisuuden tehdä korjauksia. Olettakaamme, että haluat muuttaa rivillä 40 sanan ANNA sanaksi SYÖTÄ. Korjauksen voit tehdä näin: Kirjoita

```
ED40
```

Tietokone vastaa tällöin

```
40 PRINT "ANNA TAPAHTUMA";
```

Pidä nyt kopiointinäppäintä [→] painettuna, kunnes tietokone on kirjoittanut uudestaan ohjelmarivin 40 siten, että viimeksi on mukana sana ANNA.



Poisto- ja kopiointinäppäimet

Poista nyt sana ANNA poistonäppäimellä [←] ja lisää senjälkeen kirjoittamalla sana SYÖTÄ. Muu osa rivistä pitää säilyä muuttumattomana. Sen saat aikaan pitämällä kopiointinäppäintä alhaalla, kunnes koko rivi 40 on näkyvissä kuvaruudussa. Kuten tavallista päätät nytkin rivin painamalla paluunäppäintä. Harjoittele muutamia kertoja ohjelmarivien muuttamista komennolla ED. Muista, että kaikki virheelliset merkit poistetaan poistonäppäimellä. Ennallaan säilytettävät merkit saadaan taas kuvaruutuun kopiointinäppäimellä.

Liikevaihtoveron laskeminen

Useimmat yritykset ovat velvollisia tilittämään liikevaihtoveron (lvv) käymästään kaupasta. Näille yrityksille on päivittäistä rutiinia joko laskea lvv:n määrä tai kirjata liikevaihtoveroilmoitukseen hintaan sisältyvä lvv. Jos ABC 80 -tietokonetta käytetään laskemiseen, voidaan kaikki maksut esittää kuvaruudussa samanaikaisesti. Laskennallisten virheiden riski vähenee, koska kaikki summiin liittyvät selittävät tekstit ovat koko ajan näkyvissä tarkistuksia varten. Yksinkertainen ohjelma liikevaihtoveron laskemiseksi voi näyttää seuraavalta.

Muista aina tyhjentää vanha ohjelma tietokoneen muistista käskyllä NEW!

```
10 PRINT
20 PRINT
30 PRINT "HINTA ILMAN LV-VEROA",
40 INPUT H
50 PRINT "HINTA LIIKEVAIHTOVEROINEEN",
H+H*.1628
60 PRINT "JOSTA LIIKEVAIHTOVERO ON",
H*.1628
70 GOTO 10
```

Tässä yhteydessä on huomattava, että ABC 80 jakaa kunkin rivin automaattisesti 40. merkin jälkeen. Kirjassa rivien jako on tehty mahdollisimman selkeiksi asiayhteydet huomioonottaen.

Ohjelmanajo antaa seuraavan tulostuksen, kun liikevaihtovero on 14 % tuotteen myyntihinnasta:

```
HINTA ILMAN LV-VEROA          ?200
HINTA LIIKEVAIHTOVEROINEEN     232.56
JOSTA LIIKEVAIHTOVERO ON       32.56

HINTA ILMAN LV-VEROA          ?
```

Käytä symboleita oikein

BASIC-kielessä käytetään pilkun asemesta desimaalipistettä erottamaan luvun kokonais- ja desimaaliosa toisistaan. Tämä johtuu siitä, että desimaalipiste on kansainvälisessä käytössä yleistynyt symboli. Jos kokonaisosa on nolla, tietokone kirjoittaa pelkästään desimaalipisteen ja desimaalit. Luvun $0.206\dot{3}$ tietokone kirjoittaa siten .2063. Voit myös itse käyttää tätä kirjoitustapaa syöttäessäsi lukuja tietokoneeseen. Desimaalipisteen ja kertomerkin välisten sekaannusten välttämiseksi on BASIC-kieleen valittu kertolaskun symboliksi * .

Huomaa edelleen, että meillä käytössä oleva pilkku liittyy BASIC-kielessä vain joko kirjoitettuun tulostustekstiin tai tulostuksen paikan määräämiseen.

Kuinka suurista luvuista tietokone selviää?

Kun liikevaihtovero-ohjelma ajetaan verottomalla hinnalla 1 000 000 mk, saadaan tulokseksi

```
HINTA ILMAN LV-VEROA                ?1000000
HINTA LIIKEVAIHTOVEROINEEN           1.1628E+6
JOSTA LIIKEVAIHTOVERO ON             1.628E+5
```

Tietokone työskentelee kuuden numeron sisäisellä tarkkuudella. Hyvin suurten ja hyvin pienten lukujen esittämiseksi kuvaruudussa käytetään siksi hyväksi luvun 10 potensseja. Normaalitilanteessa luku 1 206 300 kirjoitetaan luvun 10 potenssin avulla seuraavalla tavalla:

$$1{,}2063 \cdot 10^6$$

Merkinnässä 10^6 olevaa kuutosta sanotaan eksponentiksi. ABC 80:ssa ilmoitetaan luvun 10 potenssien eksponentti kirjaimen E avulla. Eksponentin edessä täytyy aina ilmoittaa etumerkki. Edellä mainittu luku ilmoitetaan ABC 80:n BASIC-kielessä seuraavasti

1.2063E+6

Tässä tapauksessa eksponentti on positiivinen. Jos eksponentti on negatiivinen, täytyy kirjaimen E perään merkitä miinusmerkki. Lukujen syötössä voidaan täysin valinnaisesti käyttää joko tavallista tai eksponenttimuotoa. Tietokone voi käsitellä huimaavan suuria lukuja. Ylärajana on 0.999999E+127 ja vastaavasti alarajana on 0.1E-127. Kokeile eksponenttimuotoa syöttämällä liikevaihtovero-ohjelmaan

1000	muodossa 1E+3
127300	muodossa 1.273E+5

Yli kuuden numeron sisältäviä lukuja käsittelemme luvussa 12 ASCII-aritmetiikan yhteydessä.

Tulos ilman desimaaleja

Kun edellä oleva ohjelma ajetaan tilanteessa, jossa ilman liikevaihtoveroa oleva hinta on 506.50, saadaan seuraava tulostus:

```
HINTA ILMAN LV-VEROA                ?506.50
HINTA LIIKEVAIHTOVEROINEEN           588.958
JOSTA LIIKEVAIHTOVERO ON             82.4582
```

Tämän esitysmuodon varjopuolena on, että tietokone kirjoittaa liian monta desimaalia. Summa 588.958 pitäisi kirjoittaa mieluummin 588.96. Seuraavana toimenpiteenä muunnamme ohjelmaa niin, että tulostamme ainoastaan täysien markkojen lukumäärän. On nimittäin mahdollista poistaa luvusta kaikki desimaalit, jolloin jäljelle jää sen "kokonaisosa".

Ohjelmarivin

```
50 PRINT "HINTA LIIKEVAIHTOVEROINEEN",
INT(H+H*.1628)
```

tietokone tulkitsee siten, että sulkeiden sisällä olevasta lausekkeen arvosta kirjoitetaan näyttöön vain kokonaisosa. Englannin sanasta INTEGER tuleva käsky lyhenne INT merkitsee kokonaislukua. INT on ns. funktio, joka vaikuttaa lukujen ja laskutoimitusten tulosten esitysmuotoon. Funktioita käsitellään lähemmin 9. luvussa. Lisää siis merkintä INT riveillä 50 ja 60 sulkeissa olevien laskutoimituslausekkeiden eteen. Ohjelma saa seuraavan muodon:

```
10 PRINT
20 PRINT
30 PRINT "HINTA ILMAN LV-VEROA",
40 INPUT H
50 PRINT "HINTA LIIKEVAIHTOVEROINEEN",
INT(H+H*.1628)
60 PRINT "JOSTA LIIKEVAIHTOVERO ON",
INT(H*.1628)

70 GOTO 10
```

Testaa ohjelma uudestaan samoilla tiedoilla eli datoilla kuin ennenkin!

```
HINTA ILMAN LV_VEROA          ?506.50
HINTA LIIKEVAIHTOVEROINEEN     588
JOSTA LIIKEVAIHTOVERO ON       82
```

Huomaat, että lasketuista luvuista puuttuvat tässä tapauksessa *pennejä merkitsevät* desimaalit. Jottei yritysten kassa esimerkiksi liikevaihtoveron laskemisen jälkeen eroaisi lasketusta oikeasta summasta, täytyy todellisessa käytännön tilanteessa ottaa huomioon myös poisjätetyt pennit jollakin sopivalla tavalla.

Laskutoimitusten järjestys

BASIC-kielessä merkitään jakolasku symbolilla /. Luvun 5 jakaminen 3:lla merkitään siis 5/3 (lukujen 5 ja 3 osamäärä). Yhdistetyissä laskutoimituslausekkeissa noudatetaan ns. AOS-logiikkaa. Sen mukaisesti suoritetaan kaikki kerto- ja jakolaskut ennen yhteen- ja vähennyslaskuja. Samanarvoiset laskutoimitukset (kuten * ja / sekä + ja —) suoritetaan vasemmalta oikealle. Huomaa, että tavanomaisessa kouluaritmetiikassa tehdään kertolaskut yleensä ennen jakolaskuja! Laskujärjestys voidaan aina muuttaa sulkeilla, jotka kumoavat kaikki tavalliset järjestys- eli prioriteettisäännöt.

Muutamia esimerkkejä laskutoimitusten kirjoitustekniikasta:

Tavanomainen kirjoitustapa	**BASIC-kielen vaatima tapa**
$5 \cdot (3+2)$ tai $5(3+2)$	5 * (3+2)
$\frac{5}{3+2}$ tai $5 : (3+2)$	5/(3+2)
$\frac{5+7}{3+2}$	(5+7)/(3+2)

Mieti, mitä allaoleva ohjelma tekee (eli suorita ns. pöytätestaus). Tarkista tulos tietokoneella.

```
10 LET A=2
20 LET B=6/2+5*A
30 LET A=(A+B)/5
40 PRINT A/3
```

Lisämahdollisuudet

Tässä luvussa suorittamissamme laskutehtävissä olemme käyttäneet hyväksi vain neljää peruslaskutoimitusta. On selvää, että tietokoneella voidaan suorittaa myös paljon monimutkaisempia ja pitemmälle kehitettyjä laskutoimituksia. Lisäksi ABC 80:n kyky graafisiin esityksiin tekee mahdolliseksi antaa tulokset hyvin havainnollisessa muodossa. Luvuissa 9 ja 10 käymme seikkaperäisesti läpi tietokoneen kyvyn käsitellä erilaisia matemaattisia ongelmia.

Muuttujatyypit (lajit)

Tässä luvussa et ole käyttänyt yhtään uutta käskyä. Olemme vain laajentaneet aikaisemmin opittujen käskyjen käyttöalueita. Käskyjä voidaan soveltaa erilaisiin muuttujatyyppeihin. ABC 80:ssa käytetään merkkijonomuuttujaa ja lukumuuttujaa. Ensin mainittua käytetään yleensä tekstien tulostamiseen ja jälkimmäistä laskutoimitusten suorittamiseen. Lukumuuttujat voidaan edelleen jakaa liukulukuja kokonaislukumuuttujiksi. Tässä luvussa käytetyt lukumuuttujat olivat liukulukumuuttujia. Kokonaislukumuuttujia käytetään erityistarkoituksiin kuten esimerkiksi suoritusnopeuden lisäämiseksi ja muistitilan säästämiseksi. Kolmen mainitun muuttujatyypin ominaisuudet selvitetään seuraavassa.

Liukulukumuuttuja

Liukulukumuuttujalla on nimenä eli symbolina kirjain tai kirjain ja numero. Se voi saada arvonsa lukujen ±0.1E-127 ja ±0.999999E+127 väliltä. Liukulukumuuttuja on aina nollattu ennen kuin sitä käytetään ensimmäistä kertaa ohjelmassa. Esimerkkejä liukulukumuuttujasta ovat A, B3, Ö9.

Kokonaislukumuuttuja

Kokonaislukumuuttuja nimetään kuten liukulukumuuttuja, mutta sen perään liitetään tunnukseksi prosenttimerkki %. Se voi saada kokonaislukuarvot lukujen —32768 ja +32768 väliltä. Kokonaislukumuuttuja on aina nollattu ennen kuin sitä käytetään ensimmäistä kertaa ohjelmassa. Esimerkkejä kokonaislukumuuttujasta ovat A%, B3%, Ö9%.

Merkkijonomuuttuja

Merkkijonomuuttuja nimetään kuten liukulukumuuttuja, mutta perään liitetään jono- eli valuuttamerkki ¤ (tai $). Jonomuuttujassa saa olla mielivaltainen määrä merkkejä, mutta jos määrä ylittää 80, täytyy käyttää erikoiskäskyä DIM, joka selvitetään myöhemmin. Esimerkkejä merkkijonomuuttujasta ovat A¤, B3¤, Ö9¤.

Yhteenveto

Käsky	Selitys	Esimerkki
PRINT	Tulostaa näyttöön. Voi sisältää myös laskutoimituslausekkeen.	10 PRINT "HEI",A$ 20 PRINT A , A% 10 PRINT (A+B)*2
LET	Sijoittaa arvon muuttujalle.	10 LET A=A*10 20 LET A%=5 30 LET A$="HEI"
	LET ABC 80:ssa voidaan jättää kirjoittamatta ABC 80 tietokoneessa.	30 A$="HEI"
INPUT	Odottaa tiedon syöttöä näppäimistöltä.	20 INPUT A 30 INPUT A% 40 INPUT A$
READ	Lukee muuttujien arvoja DATA-lauseista.	10 READ A,A%,A$
DATA	Varastoi muuttujien arvoja, jotka luetaan READ-lauseella.	20 DATA 5.31,4,HEI 30 DATA "EI SAA"
IF... THEN...	Testaa, onko looginen ilmaus tosi. Jos on, tekee sanaa THEN seuraavan lauseen.	40 IF A=5 THEN GOTO 10 50 IF A$="HEI" THEN GOTO 10

Funktio	Selitys	Esimerkki
INT(X)	Antaa suurimman kokonaisluvun, joka on pienempi tai yhtä suuri kuin X.	10 LET A=INT(6.3)

Komento	Selitys	Esimerkki
ED	Antaa mahdollisuuden ohjelmarivin muuttamiseen niin, että sitä ei tarvitse kirjoittaa uudestaan.	ED 40

Harjoituksia

4.1 Kirjoita BASIC-ohjelma, joka suorittaa seuraavat laskut:

a) (2+5)3
b) (2+5)/7
c) (a+b)ab/a kun a saa arvon 2 ja b arvon 3.

4.2 Kirjoita ohjelma, joka lukee käskyllä READ luvut 2, 4, 7, 11 ja 16 asiaankuuluvasta DATA-lauseesta ja tulostaa lukujen summan, jota edeltää teksti "SUMMA ON".

4.3 Mitä kuvaruutuun tulostuu, kun ajat seuraavan ohjelman?

```
10 LET A=5
20 LET B=A*2-8
30 LET A=B+2
40 PRINT A,B
```

4.4 Seuraava ohjelma laskee erään nelihenkisen perheen keski-iän. Perheenjäsenet ovat 36-vuotias isä Lauri, 38-vuotias äiti Anna sekä lapset Pekka 4 vuotta ja Eeva 6 vuotta.

```
10 READ L,A,P,E
20 PRINT "KESKIMÄÄRÄINEN IKÄ =";
(L+A+P+E)/4;" VUOTTA"
30 DATA 36,38,4,6
```

Muuta ohjelmaa siten, että se vastaa sinun perhettäsi.

4.5 Moottorilehti "MALTTIA LIIKENTEESEEN" on järjestänyt suuren autotestin, jossa mm. polttoaineen kulutus laskettiin tarkkaan. Jokaisesta testatusta kulkuneuvosta saatiin tiedot, kuinka monta kilometriä (S) se kulki tietyllä litramäärällä bensiiniä (L). Polttoaineen kulutus kilometriä kohden voidaan laskea jakamalla L matkalla S. Tee ohjelma, joka INPUT-lauseella annetuista testituloksista S ja L laskee kunkin ajoneuvon polttoainekulutuksen sadalla kilometrillä (l/100 km).

4.6 Yksinkertainen kertotaulu mielivaltaisille luvuille väliltä 1 ja 12 saadaan aikaan seuraavalla ohjelmalla:

Kotkan
Kaupunginkirjasto

```
10 PRINT "ILMOITA LUKU ";
20 INPUT X
30 PRINT
40 PRINT "LUVUN";X;" KERTOTAULU ON SEURAAVA:"
50 LET A=A+1
60 PRINT X;" *";A;"=";
70 IF A=12 THEN GOTO 90
80 GOTO 50
90 PRINT
```

Täydennä ohjelmaa siihen soveltuvalla laskutoimituksella riville 60.

4.7 Muuta tehtävän 4.4 ohjelmaa siten, että perheenjäsenten ikä syötetään kussakin tapauksessa sopivalla tavalla INPUT-lauseella.

KAPPALE 5

Tietokone tekee päätöksiä

ABC80

Satunnaisluku

Tietyissä yhteyksissä on toivottavaa saada käyttöön mielivaltainen luku, ns. satunnaisluku. Funktio RND tuottaa eli generoi yhden sellaisen luvun. Aja seuraava ohjelma, niin ymmärrät kuinka RND toimii.

```
10 FOR X=1 TO 60
20 PRINT RND,
30 NEXT X
```

Tulostus on seuraava:

```
.079112         .106671         .452764
.352572         .832973         .299984
.886299
j.n.e.
```

Kaikille saaduille satunnaisluvuille on selvästikin yhteistä se, että ne ovat suuruudeltaan lukujen 0 ja 1 välissä. Jos halutaan satunnaisluku rajojen 0 ja 100 välistä, muutetaan ohjelmarivi 20 muotoon

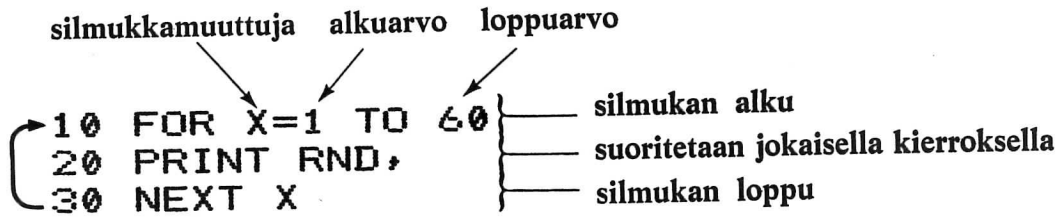
```
20 PRINT RND*100,
```

Huolimatta siitä että käsky RND esiintyy ohjelmassa vain kerran, saadaan näyttöön 60 satunnaislukua. Tämä johtuu siitä, että rivit 10 ja 30 määräävät tietokoneen suorittamaan niiden välissä olevan ohjelmanosan 60 kertaa peräkkäin. Tästä uudesta ns. silmukkamenetelmästä lähemmin seuraavassa kappaleessa.

Ohjelmasilmukat

Sellaista ohjelmanosaa, joka esiintyy ohjelmassa vain kerran, mutta joka suoritetaan useita kertoja, kutsutaan silmukaksi eli "luupiksi" (engl. loop). Ohjelmarivi 10 FOR X = 1 TO 60 määrää silmukan alun ja lopun. Muuttujaa X käytetään tässä tapauksessa ns. silmukkamuuttujana. Jokaisen kierroksen jälkeen suurennetaan silmukkamuuttujan arvoa yhdellä. Silmukkaa kierretään tällä tavoin uudestaan ja uudestaan, kunnes silmukkamuuttujan loppuarvo on saavutettu.

Silmukan loppu merkitään rivillä 30 olevalla käskyllä NEXT X. Silmukan viimeinen käsky sisältää aina silmukkamuuttujan.



Aina sama satunnaisluku?

Ohjelmaan sisältyvä funktio RND tuottaa satunnaislukuja tiettyjen matemaattisten sääntöjen mukaan. Tästä seuraa, että tietokone tuottaa normaalisti aina saman jonon satunnaislukuja. Tutki tätä ilmiötä ajamalla seuraavaa ohjelmaa muutamia kertoja.

```
10 PRINT RND,RND,RND
```

Useimmiten on kuitenkin toivottavaa, että tietokone tuottaisi aina eri satunnaislukuja eri ajokerroilla. Tämä saadaan aikaan käskyllä RANDOMIZE. Lisää rivi

```
5 RANDOMIZE
```

Nyt muodostuu selvästikin eri satunnaislukuja, kun ohjelmaa ajetaan useita kertoja. Saamme siis aikaan todellisia satunnaislukuja.

Tietokone miettii lukua

Käytämme nyt hyväksi tietokoneen kykyä tuottaa satunnaisluku. Laadimme yksinkertaisen arvausleikin, jossa tietokone generoi satunnaisluvun, tässä tapauksessa kokonaisluvun, joka on rajojen 0 ja 9 välillä (nämä mukaanluettuina). Tämä luku on kuitenkin toistaiseksi hyvin varjeltu salaisuus tietokoneen muistissa, kunnes arvaat sen lopulta oikein. Tässä on ohjelma:

```
10 PRINT
20 PRINT "ARVAA LASKEMANI
SATUNNAISLUKU ( 0 - 9 )"
30 RANDOMIZE
40 LET S=INT(RND*10)
50 PRINT "ARVAUKSESI ";
60 INPUT A
70 IF A=S THEN GOTO 110
80 PRINT "SE EI OLLUT SE OIKEA LUKU"
90 PRINT
100 GOTO 50
110 PRINT "AIVAN OIKEIN ARVATTU"
```

Myös tämä ohjelma sisältää silmukan. Ohjelmarivien 50 ja 100 rajaamaa ohjelmanosaa suoritetaan niin kauan, kunnes arvauksesi täsmää muuttujalla olevan arvon kanssa. Huomaa, että silmukkamuuttujaa ei voida käyttää hyväksi tällaisessa ohjelmassa. Onhan mahdotonta tietää etukäteen, kuinka monta arvausta tarvitset oikean luvun löytämiseksi. Kun arvauksesi täsmää tietokoneen satunnaisluvun kanssa, niin silmukasta poistutaan, mikä tapahtuu rivillä 70.

```
70 IF A=S THEN GOTO 110
```

Tietokone tekee tässä päätöksensä sen perusteella, onko arvaus A sama kuin satunnaisluku S. Jos A = S, niin tietokone siirtyy riville 110 ja kirjoittaa OIKEIN ARVATTU. Jos luvut ovat erisuuria, ei ehto ole täytetty. Tietokone jatkaa tällöin ohjelmarivien suorittamisa nousevassa järjestyksessä kuten tavallista. Rivit 80, 90 ja 100 suoritetaan, jonka jälkeen kone palaa silmukan alkuun riville 50.

Useita valinnanmahdollisuuksia

Edellisessä ohjelmassa sinun on tehtävä valintasi vain 10 luvun joukosta, joten oikean arvauksen löytäminen ei kestä kauan.

Jos sinun on sitävastoin tehtävä valintasi kaikista välillä 0—99 olevista kokonaisluvuista, tarvitset hieman apua arvauksiisi. Antakoon tietokone esimerkiksi tiedon siitä, onko ehdotettu luku liian suuri tai pieni. Tämän johtolangan avulla taitava arvaaja selviää tehtävästä aina vähemmällä kuin kahdeksalla arvauksella. Ehdotus tällaiseksi ohjelmaksi on seuraava:

```
10 PRINT
20 PRINT "ARVAA LASKEMANI
SATUNNAISLUKU ( 0 - 99 )"
30 PRINT
40 RANDOMIZE
50 LET S=INT(RND*100)
60 PRINT "ARVAUKSESI ";
70 INPUT A
80 IF A=S THEN GOTO 160
90 IF A>S THEN GOTO 130
100 PRINT "LIIAN PIENI"
110 PRINT
120 GOTO 60
130 PRINT "LIIAN SUURI"
140 PRINT
150 GOTO 60
160 PRINT "IHAN  O I K E I N  ARVATTU !"
```

Kokeile ohjelmaa ja yritä keksiä menetelmä, jolla löydät oikean luvun nopeasti!

Tässä ohjelmassa tietokoneella on valittavana kolme eri vaihtoehtoa. Arvattu luku voi olla liian suuri tai liian pieni, mutta tietysti se voi olla sama kuin tietokoneen satunnaisluku. Vaihtoehdot ovat

1. A = S
2. A > S eli A on suurempi kuin S
3. A < S eli A on pienempi kuin S

Tietokone testaa vaihtoehdon 1 ohjelmarivillä 80 ja vaihtoehdon 2 ohjelmarivillä 90. Koska on olemassa kaikkiaan vain 3 mahdollisuutta, on riittävää tutkia vaihtoehdot 1 ja 2.

Tutkikaamme, mitkä ohjelmarivit suoritetaan eri vaihtoehdoissa.

Vaihtoehto	Suoritettavat ohjelmarivit
1. A = S	10—80 sekä 160
2. A > S	10—90, 130—150, 60 jne
3. A < S	10—120, 60 jne

Ohjelman analysoinnin helpottamiseksi annetaan seuraavassa eri rivejä koskevat lyhyet kommentit.

Rivinumero	Rivin toiminta
10, 30, 110, 140	Tuottaa näyttöön tyhjän rivin.
20	Selittää ohjelman tarkoituksen.
40	RANDOMIZE varmistaa eri satunnaislukujen saamisen eri ohjelmanajoissa.
50	Sijoittaa muuttujalle kokonaislukuarvon väliltä 0—99.
60	Selittää arvaajalle, mitä pitää tehdä.
70	Muuttujalle A sijoitetaan arvoksi sinun arvauksesi.
80	Aiheuttaa hypyn riville 160, jos arvasit oikein.
90	Aiheuttaa hypyn riville 130, jos arvauksesi oli liian suuri luku.
100	Jos arvauksesi ei ollut oikea eikä liian suuri luku, suoritetaan tämä rivi, joka tulostaa "LIIAN PIENI".
120, 150	Paluuhyppy riville 60 niin, että voit arvata uudestaan.
130	Tulostaa, jos rivin 90 ehto oli täytetty.
160	Tulostaa, jos rivin 80 ehto oli täytetty.

Tietokone lajittelee lukuja

Tietyn ohjeen mukainen lukujen lajittelu on erittäin tavallinen ja sitäpaitsi helppo tehtävä tietokoneelle. Tällöin käytetään hyväksi sitä, että tietokone osaa ratkaista, kumpi annetuista kahdesta luvusta on pienempi. Tietokone osaa myös

yksinkertaisesti vaihtaa kahden luvun paikkaa. Katsokaamme ensin lähemmin, kuinka tietokone vaihtaa kahden muuttujan arvot keskenään. Olettakaamme, että muuttujalle A on annettu arvo 7 ja muuttujalle B arvo 5. Miltä näyttää ohjelma, joka vaihtaa nämä arvot keskenään? Toimisivatko seuraavat rivit?

```
10 LET A=B
20 LET B=A
```

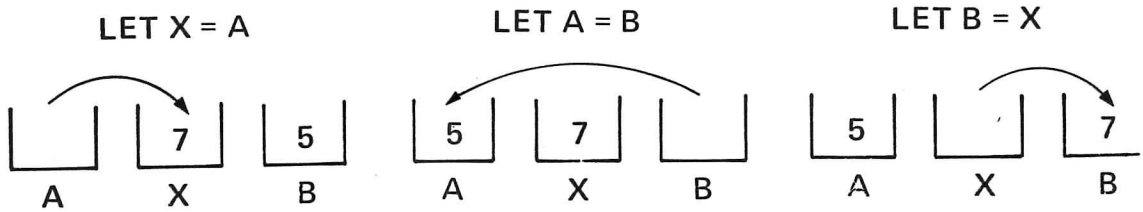
Mikseivät? Kun tietokone sijoittaa A:lle uuden arvon 5, sen aikaisempi arvo 7 häviää eikä sitä voida käyttää seuraavalla rivillä. Rivillä 20 sijoitetaan B:lle siten A:n senhetkinen eli uusi arvo, joten sekä A:lla että B:llä on ajon jälkeen sama arvo 5. Ongelman ratkaisuna on sijoittaa A:n arvo väliaikaisesti apumuuttujalle X. Tämä tapahtuu lisäämällä ohjelmarivi 5.

```
5 LET X=A
10 LET A=B
20 LET B=X
```

Kokeile asiaa ajamalla alla oleva ohjelma.

```
1 LET A=7
2 LET B=5
5 LET X=A
10 LET A=B
20 LET B=X
30 PRINT "A=";A,"B=";B
```

Tämä ohjelma toimii! Ajon jälkeen ovat muuttujien arvot A = 5 ja B = 7 eli muuttujien arvot ovat vaihtaneet paikkaa.



Apumuuttujan käyttäminen

Voimme nyt muodostaa ohjelman, joka järjestää lukuja pienimmästä suurimpaan Käytämme hyväksi tietokoneen ominaisuutta osata päätellä, kumpi kahdesta lu vusta on suurempi. Tuloksesta riippuen vaihdamme *lukujen paikat* keskenään mikäli se on tarpeen. Esitämme järjestämistekniikan esimerkin avulla.

Saakoon tietokone muuttujien A, B ja C arvot INPUT-lauseella. Nämä arvot lajitellaan ja tulostetaan pienimmästä suurimpaan. Ohjelma, joka suorittaa tämän, voi näyttää seuraavalta:

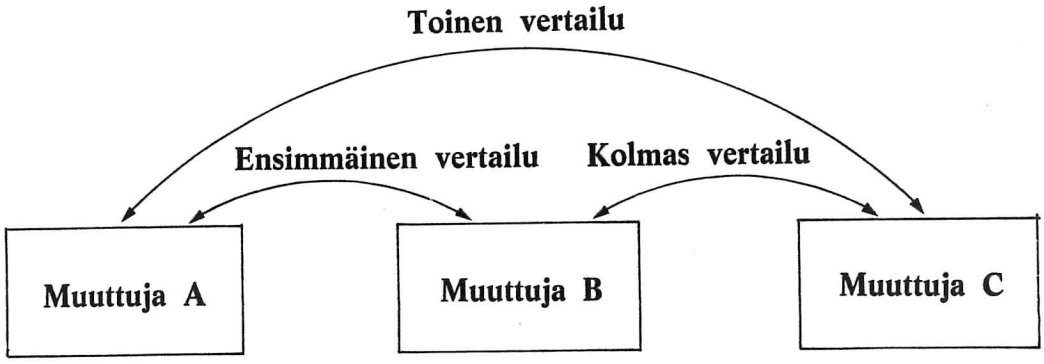
```
10 PRINT "SYÖTÄ SISÄÄN KOLME LUKUA"
20 INPUT A,B,C
30 IF A<B THEN 70
40 LET X=A
50 LET A=B
60 LET B=X
70 IF A<C THEN 110
80 LET X=A
90 LET A=C
100 LET C=X
110 IF B<C THEN 150
120 LET X=B
130 LET B=C
140 LET C=X
150 PRINT "NIIDEN SUURUUSJÄRJESTYS OLI"
160 PRINT A,B,C
```

Kokeile ohjelmaa ja koeta miettiä sen periaatetta.

Kuten varmaankin huomaat, käytämme aikaisemmin opittua menetelmää kahden muuttujan arvojen vaihtamiseksi. Toistamalla vertailuja ja vaihtoja varmistamme, että muuttujalla A on lopulta pienin syötetyistä arvoista. Tämän jälkeen vertailemme muuttujien B ja C arvoja keskenään ja tarvittaessa vaihdamme ne keskenään. Näin varmistamme, että A:lla on pienin ja C:llä suurin syötetyistä arvoista.

Rivinumero	**Rivin toiminta**
10	Selittävän tekstin kirjoittaminen.
20	Muuttujien A, B ja C arvojen syöttäminen.
30	Testataan, onko muuttujan A arvo pienempi kuin muuttujan B arvo. Myönteisessä tapauksessa ovat A:n ja B:n arvot oikeassa järjestyksessä eikä mitään vaihtoa tarvita. Tällöin hypätään riveillä 40, 50 ja 60 annetut käskyt yli.
40, 50, 60	Vaihtavat muuttujien A ja B arvot keskenään.
70	Vastaa riviä 30 muuttujille A ja C.
80, 90, 100	Vaihtavat muuttujien A ja C arvot keskenään.

110 Vastaa riviä 30 muuttujille B ja C.
120, 130, 140 Vaihtavat muuttujien B ja C arvot keskenään.
150 Selventävän tekstin kirjoittaminen.
160 Järjestetyt luvut kirjoitetaan näyttöön nousevassa järjestyksessä.



Muuttujien lajittelu

Ymmärtääksesi, miten muuttujien vaihto tapahtuu edellä olevassa ohjelmassa, voit täydentää seuraavan taulukon siten, että kullakin rivillä ovat ne muuttujan-arvot, jotka ovat voimassa, kun ko. rivi on suoritettu.

Rivinumero	Muuttujan arvo, kun ko. rivi on suoritettu			
	A	B	C	X
30	13	7	4	0
40	13	7	4	13
50	7	7	4	13
60	7	13	4	13
70	7	13	4	7
80		13	4	7
90	4	13		7
100	4	13	7	
110	4	13	4	
120	4		7	13
130	4	7		13
140	4	7	13	

Indeksoidut muuttujat

Tähän mennessä käyttämiämme liukulukumuuttujia olemme merkinneet kirjaimella tai kirjaimella ja numerolla. Kokonaislukumuuttujan merkitsemme lisäämällä muuttujaan % samoin kuin merkkijonomuuttujaan ¤ (tai $).

Joskus tarvitaan useampia muuttujia kuin on mahdollista saada aikaan tavanomaisilla muuttujanimillä. Lisäksi on usein, erityisesti tilastotehtävissä edullista ryhmitellä muuttujia jollain selkeällä tavalla. Tämän tekee mahdolliseksi ns. indeksoitu muuttuja, jossa muuttujan nimen jälkeen merkitään sulkeisiin indeksi, joka on luonnollinen luku. Seuraavassa esimerkissä on muuttujannimeen A liitetty indeksit 1:stä 5:een:

A(1), A(2), A(3), A(4), A(5)

Tällaista indeksoitujen muuttujien ryhmää kutsutaan vektoriksi eli yksiulotteiseksi taulukoksi.

Tärkeä etu indeksoidun muuttujan käytössä on se, että niiden arvojen syöttäminen on helppoa. Katsokaamme esimerkiksi, kuinka edellä olevan vektorin (taulukon) muuttujille sijoitetaan arvot.

```
10 FOR N=1 TO 5
20 INPUT A(N)
30 NEXT N
40 FOR N=1 TO 5
50 PRINT A(N),
60 NEXT N
```

Tässä ohjelmassa on silmukka, joka suorittaa viisi kierrosta, joilla annetaan arvot muuttujille A(1), A(2), A(3), A(4) ja A(5). Tämän jälkeen arvot kirjoitetaan näyttöön uuden silmukan avulla. Kokeile!

Ilman muita toimenpiteitä voit käyttää indeksejä 0—10, mutta suuremmille indekseille kuin 10 on varattava tilaa muistista. Tämä taulukon määritteleminen tapahtuu BASIC-käskyllä DIM (engl. dimensions = ulottuvuudet). Esimerkiksi

Lause	**Toiminta**
10 DIM A(100)	Varaa tilaa 101 liukuluvulle taulukkoon. Siten käytettävissä ovat indeksit 0—100.
20 DIM A%(50)	Varaa tilaa 51 kokonaisluvulle eli käytettävissä ovat indeksit 0—50.

30 DIM A ¤ (25) Varaa tilaa 26 merkkijonolle eli käytettävissä ovat indeksit 0—25.

Taulukon järjestäminen

Tarkastelemme nyt, kuinka indeksoituja muuttujia käytetään hyväksi järjestämään mielivaltaisen suuria taulukoita. Muistathan tekniikan, jolla järjestimme kolme muuttujanarvoa oikeaan järjestykseen käyttämällä parivertailuja ja arvojen vaihtoja tarvittaessa?

Jos nyt annamme neljän muuttujan A(1), A(2), A(3) ja A(4) muodostaa taulukon, niin voimme soveltaa samaa periaatetta sen järjestämiseksi. Ohjelman voimme kuitenkin tehdä lyhyemmäksi silmukoiden avulla. Olkoot indeksoiduilla muuttujilla kentässä A seuraavan taulukon mukaiset arvot:

Muuttuja	A(1)	A(2)	A(3)	A(4)
Arvo	13	11	16	7

Kuten aikaisemminkin, järjestämme muuttujanarvot nousevaan järjestykseen suuruutensa mukaan. Tämän saamme aikaan vertailemalla muuttujia pareittain ja antamalla niiden vaihtaa paikkaa tarvittaessa keskenään niin, että suurempi arvoista sijoitetaan taulukossa aina oikealle. Vertailut tapahtuvat kuten ennenkin seuraavassa järjestyksessä:

Vertailtavat arvot	**Vertailun ja mahdollisen vaihdon jälkeen**
A(1) ja A(2)	A(1) = 11, A(2) = 13, A(3) = 16, A(4) = 7 — **vaihdettu**
A(1) ja A(3)	A(1) = 11, A(2) = 13, A(3) = 16, A(4) = 7 — **ei vaihdettu**
A(1) ja A(4)	A(1) = 7, A(2) = 13, A(3) = 16, A(4) = 11 — **vaihdettu**

A(2) ja A(3)

A(1)	A(2)	A(3)	A(4)
7	13	16	11

ei vaihdettu

A(2) ja A(4)

A(1)	A(2)	A(3)	A(4)
7	11	16	13

vaihdettu

A(3) ja A(4)

A(1)	A(2)	A(3)	A(4)
7	11	13	16

vaihdettu

Yleisesti tämä vertailuperiaate voidaan kirjoittaa siten, että muuttujia A(I) ja A(J) verrataan keskenään, missä indeksi I saa arvot 1, 2 ja 3. Jokaista I:n arvoa kohti saa muuttuja J arvot I+1:stä 4:ään.

BASIC-ohjelma, joka suorittaa tämän taulukon järjestämisen, näyttää seuraavalta:

```
40 FOR I=1 TO 3
50 FOR J=I+1 TO 4
60 IF A(I)<A(J) THEN 100
70 LET X=A(I)
80 LET A(I)=A(J)
90 LET A(J)=X
100 NEXT J
110 NEXT I
```

Rivillä 60 testataan, täytyykö muuttujien arvot vaihtaa keskenään. Jos ei tarvitse, niin ohjelmassa siirrytään riville 100, jolla määrätään arvolle A(I) vertailupariksi uuden arvon A(J):lle. Jos sen sijaan vaihto on suoritettava, se tapahtuu riveillä 70, 80 ja 90. Muistathan, miksi vaihdon yhteydessä oli käytettävä apumuuttujaa X?

Ohjelmassa on kaksi (sisäkkäistä) silmukkaa, sillä I saa arvot 1, 2 ja 3, joita jokaista vastaten J saa arvot I+1 . . . 4.

Voidaksesi suorittaa ohjelman koeajon sinun täytyy lisätä rivit muuttujien arvojen syöttämiseksi sekä järjestetyn taulukon tulostamiseksi.

```
10 FOR I=1 TO 4
20 INPUT A(I)
30 NEXT I
120 PRINT "LAJITTELEN SYÖTETYT LUVUT !"
130 FOR I=1 TO 4
140 PRINT A(I);
150 NEXT I
```

Sivulla 52 annettujen arvojen mukainen ohjelman ajo antaa seuraavan tulostuksen:

```
RUN
?13
?11
?16
?7

LAJITTELEN SYÖTETYT LUVUT !

7  11  13  16

ABC80
```

Kuinka ohjelmaa on muutettava, jotta tietokone järjestäisi esim. 7 mielivaltaista lukua 4:n asemesta? Yksinkertaisesti siten, että muutat vain silmukkamuuttujien loppuarvot! Riveillä 10, 50 ja 130 muutetaan 4 luvuksi 7 ja rivillä 40 luvuksi 6.

Tämä voidaan yleistää. Jos haluat järjestää N kappaletta lukuja, niin muuta vain vastaavalla tavalla loppuarvot riveillä 10, 50 ja 130 arvoksi N ja rivillä 40 arvoksi N-1.
Muista kuitenkin, että jos käytät taulukkoa, jossa indeksi on suurempi kuin 10, täytyy käyttää DIM-käskyä. Jos esim. rivi

```
5 DIM A(100)
```

liitetään järjestämisohjelmaan, voidaan käyttää indeksiä aina arvoon 100 saakka.

Tietokone kilpailujen tuloslaskennassa

Muuan laskettelukilpailujen järjestävä urheiluseura halusi saada käyttöönsä tietokoneohjelman, joka heti kunkin osanottajan suorituksen jälkeen lajittelee senhet-

kisen tuloslistan sekä tulostaa sen TV-ruudulle. Huippukilpailussa 10 laskettelijaa pääsee mukaan loppukilpailuun. Ohjelma, joka hyödyntää niitä lajittelun periaatteita, joita juuri olemme käsitelleet, näyttää seuraavalta:

```
10 LET N=N+1
20 INPUT A(N),A¤(N)
30 FOR I=1 TO N-1
40 FOR J=I+1 TO N
50 IF A(I)<A(J) THEN 120
60 X=A(I)
70 A(I)=A(J)
80 A(J)=X
90 X¤=A¤(I)
100 A¤(I)=A¤(J)
110 A¤(J)=X¤
120 NEXT J
130 NEXT I
140 PRINT CHR¤(12)
150 PRINT N;". PARHAAN TULOS"
160 PRINT
170 FOR I=1 TO N
180 PRINT A¤(I),A(I)
190 NEXT I
200 IF N<10 THEN GOTO 10
210 PRINT "NÄMÄ OVAT LOPPUTULOKSET"
```

Rivinumero	Rivin toiminta
10	Muuttuja N kasvaa yhdellä jokaisen lajittelun jälkeen eli se huolehtii kilpailuosuutensa jo suorittaneiden lukumäärästä.
20	Muuttujalle A(N) syötetään kunkin kilpailijan suoritusaika ja merkkijonomuuttujalle A¤(N) hänen nimensä.
40—130	Kaksoissilmukassa, jossa lajittelu tapahtuu, kuten ohjelmalistauksesta nähdään, on meidän vaihdettava arvoja sekä taulukossa A että taulukossa A¤. Muuten lajittelutekniikka on kuten aikaisemmin käsiteltiin. Pikku hienoutena — kuten itse voit havaita — on käsky LET jätetty pois. Tämä on aina sallittua ABC 80 tietokoneessa kirjoitustyön helpottamiseksi.
140	CHR¤(12), joka esiintyy käskyn PRINT jäljessä, on kuvaruudun tyhjennyskäsky. Seuraava tulostus tämän käs-

kyn suorituksen jälkeen alkaa kuvaruudun vasemmasta yläkulmasta.

150 Tilannetta selventää teksti.

160 Aiheuttaa pelkän rivinsiirron.

170—1280 Silmukka, jossa tulostetaan lajiteltu tuloslista.

Kokeile ohjelmaa syöttämällä esimerkiksi seuraavat erään kansainvälisen kilpailun ajat ja nimet. Syötä ensin kunkin kilpailijan aika, pilkkumerkki (,) ja sitten kilpailijan nimi, sillä rivin 20 INPUT-lauseessa on kaksi muuttujaa.

Aika	**Nimi**
100-34	STENSTARK
103.41	TÖNTI
100.92	HAJDHÄGGER
102.31	PILNER
100.25	HÄGGINEN
102.31	NÖJRÖJTER
101.19	LABBALAINEN
106.37	LANGENNUT
102.35	MYÖTÄMÄKI
105.83	GRÖSS

Yhteenveto

Indeksoitua muuttujaa, jossa on yksi indeksi, kutsutaan yksiulotteiseksi taulukoksi eli vektoriksi. Muuttujalla voi olla myös useampia indeksejä, jolloin muuttujasta käytetään yleisnimitystä kenttä. Erityisesti kaksiulotteisesta indeksoidusta muuttujaryhmästä eli taulukosta käytetään nimitystä matriisi. Niitä käsittelemme 14. luvussa. Indeksinä voidaan aina käyttää kokonaislukuja 0—10, ellei kenttää laajenneta käskyllä DIM.

IF-lauseissa käytetään ns. loogista ilmausta. Looginen ilmaus on aina tosi tai epätosi. Jos ilmaus on tosi, suoritetaan sanaa THEN välittömästi seuraava käsky. Se voi olla mikä tahansa BASIC-käsky. Jos se erityisesti on GOTO, niin tämä voidaan korvata pelkällä rivinumerolla. Jos ilmaus on epätosi, suoritetaan ohjelman seuraava rivi.

Looginen operaattori	Selitys	Esimerkki
>	Suurempi kuin. Muistisääntö: "Suurempi nokkii pienempää".	`10 IF A>B THEN GOTO 50`
<	Pienempi kuin.	`20 IF A<B THEN 50`
=	Yhtä suuri kuin.	`30 IF A=B THEN PRINT A`
>=	Suurempi tai yhtä suuri kuin	`40 IF A>=B THEN 10`
<=	Pienempi tai yhtä suuri kuin.	`50 IF A<=B THEN 70`
<>	Ei yhtä suuri kuin eli erisuuri kuin.	`60 IF A<>B THEN GOTO 10`

Käsky	Selitys	Esimerkki
RANDOMIZE	Antaa RND-toiminnalle umpimähkäisen alkuarvon.	`10 RANDOMIZE`
FOR... TO...	Suorittaa ohjelmanosan niin monta kertaa kuin silmukkamuuttujan alku- ja loppuarvoilla määrätään.	`20 FOR X=1 TO 5`
NEXT	FOR-silmukan päätekohta.	`50 NEXT X`

LET	Sijoittaa muuttujalle arvon. LET voidaan myös jättää kirjoittamatta.	`30 LET A=5`
		`30 A=5`
		`40 LET A$=B$`
		`40 A$=B$`

DIM	Varaa muistitilaa indeksoidulle muuttujalle (taulukolle).	`10 DIM A(50)`
		`20 DIM A%(75)`
		`30 DIM A$(30)`

Toiminta	Selitys	Esimerkki
RND	Tuottaa satunnaisluvun väliltä 0—0.999999.	`10 LET A=RND` `20 LET B=RND*10`

Erikoismerkki	Esimerkki	Selitys
CHR$(12)	Tyhjentää kuvaruudun ja siirtää osoittimen sen vasempaan yläkulmaan.	10 PRINT CHR$(12)

Harjoituksia

5.1 Ensimmäisen luokan oppilaille annettiin lähes mahdottomalta tuntuva tehtävä kirjoittaa 20 kertaa peräkkäin lause

```
ÄITI SOUTAA JA ISÄ VETÄÄ UISTINTA
```

Kirjoita ohjelma, joka suorittaa tämän tehtävän.

5.2 Urheiluseura AUS myi pikkujoulujuhlissaaan 100 arpaa, joiden numerot olivat 0—99. Voittoja oli 10. Tee ohjelma, joka arpoo voittonumerot ja tulostaa voittolistan seuraavasti:

```
VOITTOLISTA

VOITTO 1      NUMERO...
VOITTO 2      NUMERO...

jne.
```

5.3 Muuta tehtävän 5.2 ohjelmaa siten, että sitä voidaan käyttää myös seuran Uudenvuodenjuhlissa, joissa arpoja myydään 200 kpl, jotka on numeroitu 100—299. Voittoja on 15 kpl.

5.4 Kirjoita ohjelma, joka simuloi nopanheittoa tulostaen aina heiton jälkeen

```
NOPPA PYSÄHTYI , TULOS ON...
```

5.5 Kirjoita ohjelma, jonka avulla voit harjoitella päässälaskun jaloa taitoa luvuilla, jotka ovat pienempiä kuin 100. Siis, tulosta kuvaruutuun kaksi satunnaista lukua, jotka molemmat ovat välillä 0—100. Vastaus syötetään INPUT-lauseella ja sen tarkistaa tietokone. Se sitten ilmoittaa, oliko vastaus oikein vai väärin. Mikäli se oli jostakin syystä väärin, kysymys tulostetaan uutta yritystä varten.

5.6 10 matematiikan ammattilaista suoritti tehtävän 5.5. Kukin heistä ajoi sen 50 kertaa peräkkäin. Allaoleva lista kertoo, kuinka monta oikeata vastausta he saivat ensimmäisellä yrityksellä.

Koehenkilö Nr	Oikeita vastauksia 1.:llä yrityksellä
1	26
2	41
3	29
4	12
5	33
6	48
7	42
8	19
9	13
10	38

Syötä ylläolevat tulokset yksiulotteisena kenttänä ja laske kymmenen osallistujan ensimmäisellä kerralla oikein vastattujen kysymysten keskiarvo. Keskiarvon saat laskemalla yhteen tulokset ja jakamalla summan osallistujien lukumäärällä. Tulosta lopuksi osallistujien tulokset sekä laskettu keskiarvo.

5.7 Muunna arvausohjelmaa sivulla 46 siten, että se tulostaa myös, kuinka monta yritystä oikean vastauksen löytäminen vaati.

5.8 Seuraava ohjelma tulostaa veikkausrivin:

```
10 PRINT CHR$(12)
20 RANDOMIZE
30 PRINT "VEIKKAUSRIVI"
40 PRINT
50 FOR X=1 TO 13
60 PRINT "OTTELU No";X,
70 T=RND
80 IF T<.6 THEN 130
90 IF T<.8 THEN 150
100 PRINT "      2"
110 NEXT X
120 GOTO 170
130 PRINT " 1"
140 GOTO 110
150 PRINT "    X"
160 GOTO 110
170 PRINT
180 PRINT "ONNEA YRITYKSELLESI"
```

Mikä on todennäköisyys ykkösille, risteille ja kakkosille? Muunna ohjelmaa siten, että todennäköisyys ykkösille on 50 %, risteille 25 % ja kakkosille samoin 25 %.

5.9 Salaisten agenttien pääsykokeessa EkOAtK:ssa (erittäin salainen koodi) suoritetuissa testeissä todettiin eri hakijoilla seuraavat älykkyysosamäärät (ÄO):

Hakija Nr	ÄO
001	36
002	41
003	59
004	110
005	74
006	107
007	130

Kirjoita ohjelma, joka ilmoittaa, kuka hakijoista asetetaan ensisijalle ÄO:n perusteella. Loput hakijat on järjestettävä heidän ÄO:nsa mukaiseen varalistaan.

KAPPALE 6

Tietokone tallentaa ohjelmia

ABC80

Nopea ohjelmien syöttö

ABC 80 on hyvin helppokäyttöinen tietokone, mutta silti vie paljon aikaa kirjoittaa näppäimistöllä pitkiä BASIC-ohjelmia. Kasettiaseman avulla voidaan ohjelmointia kuitenkin rationalisoida huomattavasti. Kun ohjelma on valmis, se voidaan tallentaa kasetin magneettinauhalle siten, että annetaan tietokoneelle käsky tallentaa ohjelma kasetille. Jos haluat myöhemmin käyttää samaa ohjelmaa, sinun tarvitsee vain panna kasetti nauhuriin ja antaa tietokoneelle määräys hakea haluttu ohjelma.

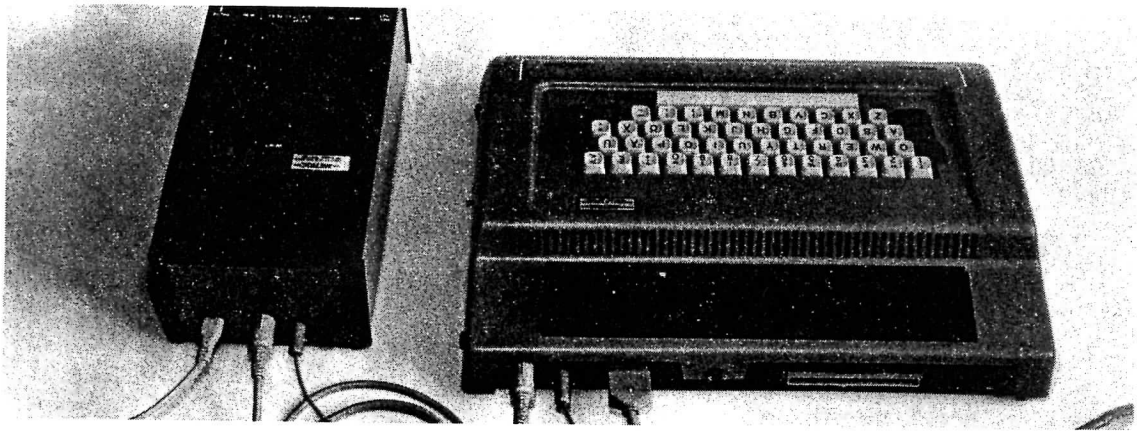
Ohjelma siirtyy tietokoneen ja kasettiaseman välillä hyvin nopeasti; nopeus on noin 70 merkkiä sekunnissa. Nopea konekirjoittaja selviää nopeudessa noin kymmenesosaan tästä ja keskimääräinen ABC 80:n käyttäjä on ymmärrettävästi vieläkin hitaampi, koska ohjelma pitää kirjoittaa virheettömästi käsin näppäimistön avulla. Kasettien käyttö auttaa siten voittamaan paljon aikaa. Katsokaamme nyt tämä käytännössä.

Kasettiaseman kytkeminen

ABC 80 -järjestelmään kuuluu kasettiasema, joka on varustettu siten, että se voidaan käynnistää tai pysäyttää tietokoneen antamilla signaaleilla. Tietokoneen ohjaussignaaleista riippumatta voidaan nauhaa pikakelata molempiin suuntiin.

Kun etsit ohjelmaa kasetilta, on paras apuneuvo laskuri. Voit myös helposti löytää ohjelman kuuntelemalla nauhoitusta. Lisäksi voit käyttää pikakelausta nauhan pyöriessäkin ilman nauhan pysäyttämistä. Nauhalla on tosin taipumus venyä ajan mittaan, jolloin sillä oleva tieto vääristyy. Varo siis todella tärkeitä kasetteja.

Tavanomaisen nauhuriliitännän lisäksi tuleva moottoriohjaus merkitsee sitä, että tarvitsemme kaksi liitäntäjohtoa näppäimistön ja kasettiaseman yhteenkytkemiseksi. Kytkentä selviää seuraavasta kuvasta.



Näppäimistön ja kasettiaseman yhteenkytkentä

Nauha

Ohjelmien tallentamiseen käytettäviltä kaseteilta vaaditaan erittäin hyvää laatua, jotta estetään ns. drop out. Tällä tarkoitetaan sitä, että nauhan oksidikerroksessa oleva epätasaisuus aiheuttaa merkin katoamisen (pois putoamisen). Tällä on tietenkin monia haitallisia seurauksia, ja sen vuoksi on syytä valita tunnettujen valmistajien hyviä kasetteja. Toimi lisäksi huolellisesti: kasettejasi voivat vaurioittaa lika ja korkea lämpötila. Lisäksi voi sattua mekaanisia vaurioita. Älä esimerkiksi koskaan käytä pikakelausta nauhoittimen PLAY-näppäimen ollessa alaspainettuna.

Ohjelmien tallennus

Kasetissa oleva nauha on yleensä varustettu alussa ja lopussa olevilla suojanauhoilla. Niinpä on syytä aluksi pyörittää kasetin keloja esimerkiksi lyijykynällä niin, että alkunauha ei ole enää näkyvissä. Tosin ABC 80 aloittaa nauhoituksen vasta noin viiden sekunnin kuluttua käynnistyksestä, mutta jos kasetin suojanauha on kovin pitkä, saattaa esiintyä häiriöitä.

Liitä kasettiasema näppäimistöön ja pistorasiaan, kytke jännite, sijoita kasetti paikalleen ja nollaa nauhalaskuri. Näytämme nyt, kuinka voit käyttää kasettiasemaa. Esimerkeiksi valitsemme joitain lyhyitä ohjelmia, joissa kussakin on vain yksi rivi.

Syötä näppäimistöllä tietokoneeseen ensimmäinen ohjelma

```
10 PRINT "OHJELMA1 ON SUORITETTU"
```

Saata nauhuri nauhoitusasentoon näppäilemällä alas kumpikin näppäimistä REC ja PLAY. Tallenna ohjelma kasetille komennolla SAVE.

Kirjoita esimerkiksi

```
SAVE OHJELMA1
```

Älä unohda päättää toimintakäskyä näppäilemällä RETURN. Tietokone käynnistää nyt nauhurin. Kun näyttöön ilmestyy ABC 80 noin 13 sekunnin kuluttua, ohjelma on nauhoitettu.

Poista rivi 10 ja syötä tietokoneeseen näppäimistöllä seuraava ohjelma:

```
20 PRINT "NYT ON MYöS OHJELMA2 AJETTU"
```

Tallenna tämäkin ohjelma kasetille kirjoittamalla

```
SAVE OHJELMA2
```

Olemme valinneet ohjelmille nimet OHJELMA 1 ja OHJELMA 2. Suurin mahdollinen merkkien määrä ohjelman nimessä on 12. Jos ohjelman nimi on pitempi kuin kahdeksan merkkiä, täytyy yhdeksännen merkin olla piste. Vältä kuitenkin vielä tässä vaiheessa pitempien kuin 8 merkin nimiä.

Kelaa nyt nauha alkuun ja paina nauhurin näppäin PLAY alas.

Aja ensimmäinen ohjelma komennolla

```
RUN OHJELMA1
```

Noin puolen minuutin kuluttua on tietokone hakenut ja ajanut ohjelman. Varmistu, että ohjelma on tietokoneen muistissa, käyttämällä käskyä LIST tavanomaiseen tapaan.

Seuraava ohjelma, OHJELMA 2, voidaan nyt ajaa ilman edeltävää takaisinkelausta. Kirjoita

```
RUN OHJELMA2
```

Samalla kun tämä ohjelma siirtyi nauhalta tietokoneen muistiin, sieltä poistui OHJELMA 1. Tämän voit helposti todeta käskyllä LIST.

Ei ole välttämätöntä tietää ohjelman nimeä, jotta sen voisi siirtää nauhalta tietokoneen muistiin.

Kelaa nauha takaisin alkuun ja kokeile komentoa

```
RUN CAS:
```

Tietokone käynnistää nauhurin ja noutaa sen ohjelman, jonka alku ensimmäisenä tulee nauhalla vastaan sekä tulostaa kuvaruutuun

```
FOUND OHJELMA1.BAC
```

Koko ohjelman siirryttyä muistiin, tietokone aloittaa sen suorituksen. Paitsi ohjelman nimen sisältämät (enintään kahdeksan) merkit, liittyy nimeen vielä neljä lisämerkkiä lähinnä ohjelmatyypin ilmaisemista varten. Niistä ensimmäinen on aina piste (.). Mikäli käyttäjä ei itse hyödynnä näitä merkkejä, lisää tietokone automaattisesti tiedon .BAC annettuun ohjelmanimeen.

Mikäli et toivo, että nauhalta tuleva ohjelma ajetaan välittömästi, vaan haluat vain siirtää sen tietokoneen muistiin, voit vaihtaa edellisen esimerkin komennon RUN komentoon LOAD. Kelaa nauha alkuun ja kirjoita

```
LOAD CAS:
```

Nyt voit tutkia nauhalta siirtynyttä ohjelmaa komennolla LIST, ennenkuin ajat sen. Ohjelman suoritus aloitetaan kuten tavallisesti komennolla RUN.

Toisinaan olisi käytännöllistä liittää nauhalla oleva toinen ohjelma jo tietokoneen muistissa olevaan ohjelmaan tai sen osaan tuhoamatta sitä. Vanha ohjelmahan poistuu kokonaan muistista noudettaessa nauhalta uusi ohjelma komennolla RUN tai LOAD. Komento uuden ohjelman liittämiseksi vanhaan on

```
MERGE CAS:
```

Mikäli tietokoneen on itse etsittävä tietyn niminen ohjelma nauhalta ja lisättävä se siihen ohjelmaan, joka jo on sen muistissa, on sinun käytettävä komentoa MERGE, jonka jälkeen annat etsityn ohjelman nimen, esimerkiksi MERGE OHJELMA 2.

Tähän asti kuvatuissa menetelmissä sinun oli itse annettava tietokoneelle komento noutaa ohjelma nauhalta. Voit kuitenkin myös antaa tietokoneelle itselleen ohjeet hoitaa ohjelmanvaihto täysin omin päin kirjoittamalla ohjelmaan käsky tätä tarkoitusta varten.

Tallenna seuraavat kaksi ohjelmaa nauhalle. Kutsu ensimmäistä nimellä OHJ1 ja toista nimellä OHJ2.

```
10 PRINT "OHJELMA1 ON SUORITETTU"
20 CHAIN "OHJ2"
```

Tallenna ohjelma käskyllä SAVE OHJ1.

Kirjoita rivi 10 uudelleen seuraavasti:

```
10 PRINT "OHJELMA2 ON SUORITETTU"
```

Tallenna ohjelma käskyllä SAVE OHJ2. Kelaa sitten nauha takaisin ja kirjoita

```
RUN CAS:
```

Molemmat ohjelmat tulevat tällöin ajetuksi vuoronperään. Kun käsky CHAIN "OHJ2" suoritetaan ensimmäisessä ohjelmassa, niin tietokone käynnistää nauhurin ja siirtää ohjelman OHJ2 kasetilta tietokoneen muistiin. Tällä tavalla voidaan tietokone saada suorittamaan niinkin pitkiä ohjelmia, jotka eivät mahdu sen muistiin yhdellä kerralla.

Tietolevyasema tallennusvälineenä

Ohjelmien tallennusta varten voit liittää ABC 80 -tietokoneeseen myös tietolevyaseman. Tietolevyn edut korostuvat, kun on tallennettava laajoja ohjelmia tai suuria tietorekistereitä. Koko tietokoneen muistin täyttävän ohjelman tallentaminen kasetille kestää noin 4 minuuttia. Saman ohjelman siirtäminen ABC 80:sta tietolevylle vie vain muutamia sekunteja kaikkine tarkistuksineenkin.

Yhteenveto

Käsky	Selitys	Esimerkki
CHAIN	Hakee ilmoitetun ohjelman kasetilta (ns. ketjutus).	100 CHAIN "OHJ2"

Komento	Selitys	Esimerkki
SAVE	Tallentaa nimetyn ohjelman kasetille.	SAVE OHJ1
RUN ...	Hakee kasetilta nimetyn ohjelman ja ajaa sen.	RUN OHJ2
RUN CAS:	Hakee kasetilta lähinnä seuraavan ohjelman ja ajaa sen.	RUN CAS:
LOAD ...	Siirtää nimetyn ohjelman kasetilta muistiin.	LOAD OHJELMA
LOAD CAS:	Siirtää lähinnä seuraavan ohjelman kasetilta muistiin.	LOAD CAS:
MERGE ...	Hakee nimetyn ohjelman kasetilta ja siirtää sen tietokoneen muistiin poistamatta sen aikaisempaa sisältöä. Jos sekä kasetilla että muistissa on sama rivinumero, tulee kasetin rivi käyttöön.	MERGE OHJELMA3
MERGE CAS:	Hakee lähinnä seuraavan ohjelman kasetilta tyhjentämättä muistia. Jos sama rivinumero esiintyy sekä kasetilla että muistissa, tulee kasetin rivi käyttöön.	MERGE CAS:

KAPPALE 7

Lohkokaaviot ohjelmoinnin punaisena lankana

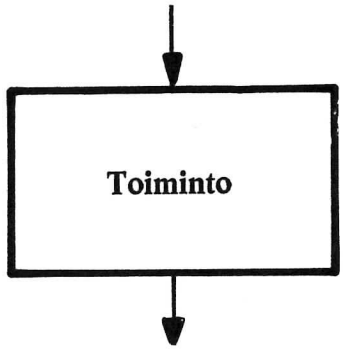
ABC80

Vuokaavio ohjelmoinnin apukeinona

Tähän mennessä on ollut suhteellisen helppoa kirjoittaa ohjelmat niihin tehtäviin, jotka meidän on ollut suoritettava tietokoneen avulla. Lyhyen ohjelman tarkistaminen on helppoa, mutta kun ohjelma käsittää enemmän kuin 20—30 riviä, voi olla vaikeaa konstruoida virheetön ohjelma. On kuitenkin olemassa hyviä apukeinoja ohjelmointitehtävien ratkaisemiseksi. Yksi parhaista on vuokaavio.

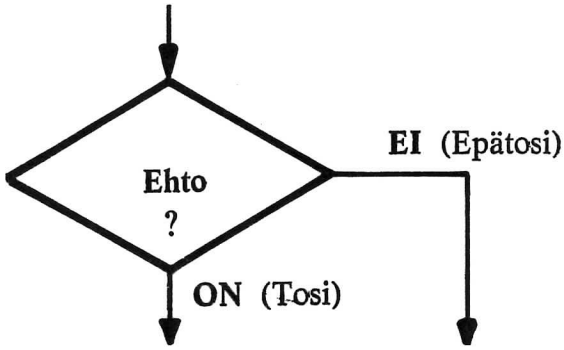
Vuokaavion peruslohkot

Yksinkertaisen vuokaavion laatimiseksi vaaditaan vain kaksi peruselementtiä; toimintalohko ja päättelylohko eli ehtolohko. Toimintalohko määrää, että sen sisältämä toiminto suoritetaan.



Toimintolohko

Päättelylohko sisältää sen, että tietokone tutkii, onko lohkon sisältämä ehto täytetty vai ei.



Päättelylohko

Päättelylohkosta on kaksi ulospääsyä. Tietokone tulee jatkamaan ohjelman suoritusta ohjelman eri riveiltä sen mukaan, onko ehto täytetty vai ei.

Vuokaavion laatiminen

Kuvatkaamme nyt tietokoneen tehtävää vuokaaviolla. Olkoon tehtävänä laskea, kuinka pääoma muuttuu vuosi vuodelta koron vaikutuksesta.

Kirjoitamme ensin sen, mitä tehtävässä on suoritettava oikeassa järjestyksessä ja kohta kohdalta.

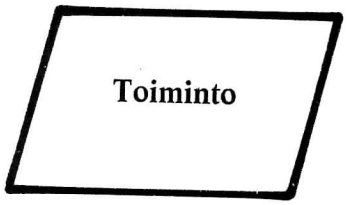
1. Pääoman suuruus, K, määritellään.
2. Korkoprosentti, P, ilmoitetaan.
3. Annetaan laskemista koskevien vuosien lukumäärä T.
4. Lasketaan yhden vuoden muutosta vastaava uusi pääoma: $K = K + K \cdot P/100$
5. Annetaan näyttöön liittyvä teksti:
   PÄÄOMA, KORKO-%, VUOSI
6. Lasketut ja annetut arvot esitetään taulukkona: K, P, T
7. Palataan kohtaan 4, jos läskut eivät ole loppuun suoritetut
8. Loppu.

Tehtävä voidaan ratkaista tällä tavalla. Valitettavasti taulukon otsikko PÄÄOMA, KORKO-%, VUOSI tulee kirjoitetuksi näyttöön joka kerta, kun uusi pääoman arvo lasketaan. Kohtien 4 ja 7 välinen osa suoritetaan monta kertaa, mikä merkitsee silmukkaa valmiissa ohjelmassa.

Välttääksemme otsikkotekstin kirjoittamisen jokaisen uuden laskemisen yhteydessä asetamme kohdaksi 4a ehdon:

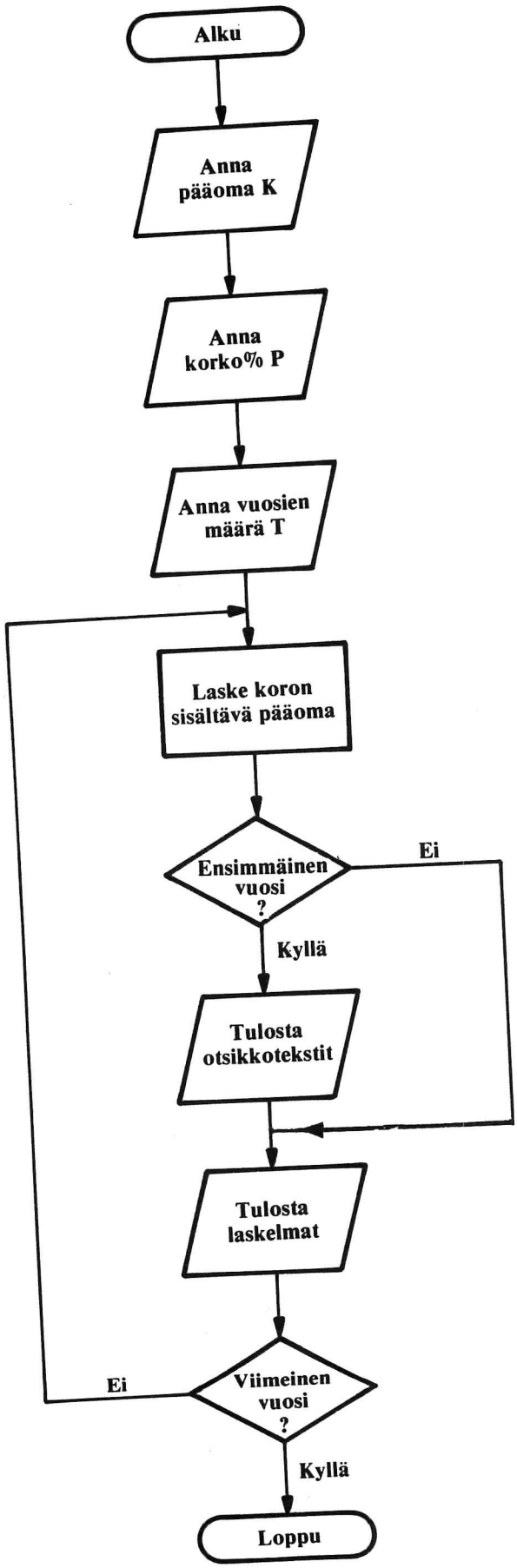
4a Jos ensimmäisen vuoden pääoma on laskettu, jätetään kohta 5 suorittamatta eli se hypätään yli.

Edellä esitettyjen yhdeksän kohdan liittäminen toisiinsa antaa seuraavalla sivulla esitetyn vuokaavion. Siinä esiintyviä symboleja tulee verrata valmiin ohjelman riveihin. Tässä kaaviossa esiintyy mm. seuraava symboli



Syöttö tai tulostus

Sitä käytetään vain määrätyissä tehtävissä, joita esiintyy joko näyttöön saatavissa tulostuksissa tai näppäimistöllä suoritettavissa syötöissä. Vuokaaviosymbolien luettelo annetaan tämän luvun lopussa.


Alku
Anna pääoma K
Anna korko% P
Anna vuosien määrä T
Laske koron sisältävä pääoma
Ensimmäinen vuosi ?
Ei
Kyllä
Tulosta otsikkotekstit
Tulosta laskelmat
Viimeinen vuosi ?
Ei
Kyllä
Loppu

Vuokaavion jokainen symboli kirjoitetaan nyt ohjelman osaksi. Askelittain saamme

1.
```
10 PRINT "ILMOITA PÄÄOMA";
20 INPUT K
```

2.
```
30 PRINT "ANNA KORKOPROSENTTI";
40 INPUT P
```

3.
```
50 PRINT "SYÖTÄ VUOSIEN MÄÄRÄ";
60 INPUT T
70 FOR N=1 TO T
```

4.
```
80 LET K=K+K*P/100
```

4a.
```
90 IF T>1 THEN [          ]
```

Tästä puuttuu rivinumero. Kun koko vuokaavio on käyty läpi, täytät tähän oikean rivinumeron.

5.
```
100 PRINT
110 PRINT "PÄÄOMA","KORKO-%","VUOSI"
120 PRINT
```

6.
```
130 PRINT K,P,N
```

7.
```
140 NEXT N
```

8.
```
150 END
```

Riviltä 90 puuttuva rivinumero on selvästi 130. Ohjelmariviksi 90 tulee siten

```
90 IF T>1 THEN 130
```

Käsky END rivillä 150 ilmoittaa, että ohjelma loppuu tässä. Tähän mennessä emme ole käyttäneet käskyä END, eikä sitä useimmiten tarvitakaan. Palaamme asiaan myöhemmin tässä luvussa.

Aja nyt ohjelma esimerkiksi 900 mk:n pääomalla ja 7 % korolla. Valitse ajaksi 18 vuotta. Tulokseksi saat 3041,97 mk.

Näppäimistö ohjelmanajon säätelijänä

Jos haluamme laskea pääoman muutokset korkoa korolle menetelmällä esimerkiksi 50 vuoden aikana, ei ohjelmamme ole nykymuodossaan oikein käyttökelpoinen. Tietokoneen nopeudesta johtuu, että kaikki 50 laskutoimitusta tapahtuvat hyvin nopeasti ja näyttöön jäävät vain 21 viimeistä vuotta.

Jos ohjelma sen sijaan kirjoitetaan siten, että laskutoimitukset suoritetaan päättymättömässä silmukassa niin, että joka kierrokseen sisällytetään tilapäinen pysähdys, voidaan tulostusta lisätä näytössä vuosi vuodelta. Ohjelmanajo voidaan tietysti keskeyttää INPUT-lauseella, mutta vielä käytännöllisempää on sen sijasta käyttää GET-lausetta.

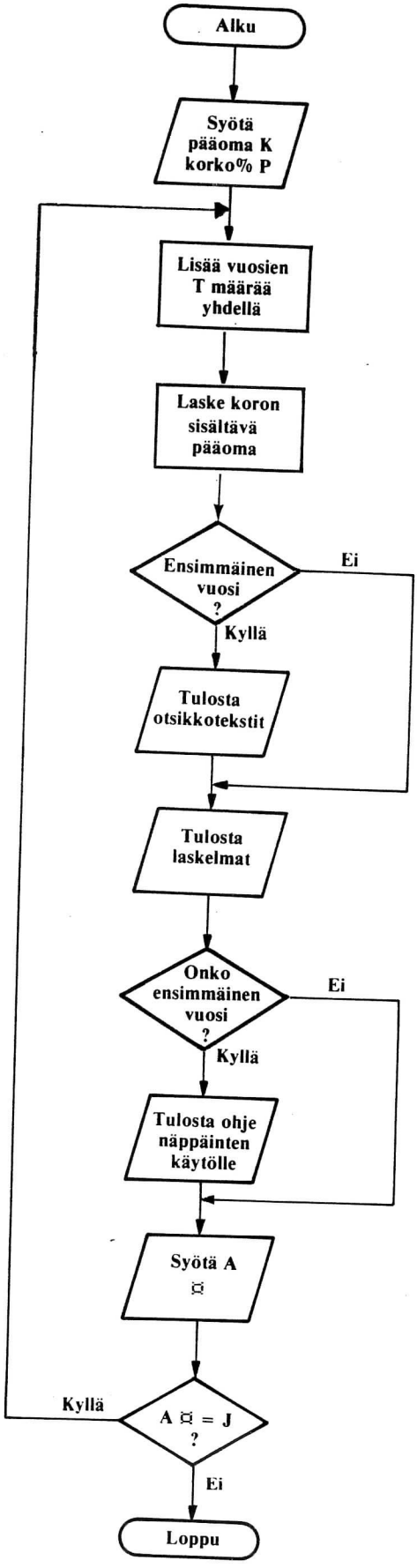
Rivi

```
160 GET A¤
```

merkitsee sitä, että muuttujalle A¤ sijoitetaan arvoksi yksi ainoa merkki. Tätä merkkiä ei kirjoiteta näyttöön. Sitäpaitsi GET-lause on sikäli erikoinen, että näppäintä RETURN ei tarvitse painaa syötön lopuksi. Tietokone jatkaa ohjelmanajoa välittömästi, kun jotain näppäintä on painettu.

Uuden ohjelman vuokaavio on seuraavalla sivulla. Uudessa ohjelmassa voidaan aikaisemman ohjelman useimmat rivit säilyttää ennallaan.

```
10 PRINT "ILMOITA PÄÄOMA";
20 INPUT K
30 PRINT "ANNA KORKOPROSENTTI";
40 INPUT P
50 }
60 } Huom! Rivit 50 ja 60 on poistettava.
70 LET T=T+1
80 LET K=K+K*P/100
90 IF T>1 THEN 130
100 PRINT
110 PRINT "PÄÄOMA","KORKO-%","VUOSI"
120 PRINT
130 PRINT K,P,T
140 IF T>1 THEN 160
150 PRINT " #JOKA KERTA KUN PAINAT
NÄPPÄINTÄ J  #   #JATKUU LASKENTA
VUODEN KERRALLAAN  #"
160 GET A¤
170 IF A¤="J" THEN 70
```


Alku
Syötä pääoma K korko% P
Lisää vuosien T määrää yhdellä
Laske koron sisältävä pääoma
Ensimmäinen vuosi ?
Ei
Kyllä
Tulosta otsikkotekstit
Tulosta laskelmat
Onko ensimmäinen vuosi ?
Ei
Kyllä
Tulosta ohje näppäinten käytölle
Syötä A ¤
Kyllä
A ¤ = J ?
Ei
Loppu

Kokeile ohjelmaa antamalla tietokoneen laskea pankkisaldo vuonna 1979, jos yksi markka olisi talletettu vuonna 1900 ja korko olisi ollut koko ajan 7 %. Tulokseksi saadaan 209,57 mk (näytössä 209,565 mk).

Jos pidät näppäintä J alaspainettuna koko ajan, ohjelmanajo käy hieman nopeammin. Kuten aikaisemmin on mainittu ovat kaikki näppäimet toistavia.

Viive ohjelman suorituksessa

Kun tietokone joutuu esittämään tekstiä enemmän kuin näyttöön kerralla mahtuu, voidaan käyttäjän antaa edellisen esimerkin mukaisesti näppäillä mielivaltaista näppäintä, jolloin tietokone jatkaa tekstin esittämistä ohjelman mukaisesti. Joskus on sopivampi vaihtoehto antaa tietokoneen odottaa tietty aika automaattisesti ja jatkaa sen jälkeen ohjelmanajoa. Tällainen viive saadaan helposti aikaan laatimalla silmukka, joka käydään läpi sopivan monta kertaa. Kokeile ohjelmaa

```
10 FOR V=1 TO 2000
20 NEXT V
30 PRINT "VIIVÄSTYS eli VIIVE"
```

Tämä viive kestää noin kaksi sekuntia. Kaksinkertaista viiveen kesto vaihtamalla rivin 10 käsky seuraavaksi ja aja ohjelma uudestaan.

```
10 FOR V=1 TO 4000
```

Voit todeta esimerkiksi kellon avulla viiveen kaksinkertaistuneen.

Useita lauseita samalla ohjelmarivillä

ABC 80 -tietokoneen BASIC-kielessä on mahdollisuus kirjoittaa useita lauseita samalle riville, kunhan ne erotetaan toisistaan kaksoispisteellä. Edellä oleva ohjelma saa uudelleen kirjoitettuna seuraavan muodon:

```
10 FOR V=1 TO 4000 : NEXT V
30 PRINT "VIIVÄSTYS eli VIIVE"
```

Tästä on joskus hyötyä pitkien ohjelmien teossa, sillä menetelmä säästää jonkin verran koneen muistitilan käyttöä. Ole kuitenkin tarkkana, sillä hypyt voivat tapahtua vain numeroidun rivin alkuun, ei lauserivin keskelle.

Lyhyempiä ohjelmia aliohjelmien avulla

Kirjoittakaamme nyt ohjelma, joka näyttää periaatteen sille, kuinka tekstiä voidaan asteittain tulostaa näyttöön. Olkoon tietokoneen tehtävänä ilmoittaa elokuvateatterissa esitettävät elokuvat viikon aikana. Jokaisen elokuvan tiedot esi-

tetään kolmen sekunnin ajan, jonka jälkeen ilmoitetaan seuraava elokuva jne., kunnes kaikki elokuvat on ilmoitettu. Seuraavalla sivulla on vuokaavio, joka ilmoittaa, mikä teksti kulloinkin on näytössä. Toiminto ODOTA 3 s esiintyy neljä kertaa vuokaaviossa. Siitä huolimatta meidän tarvitsee kirjoittaa tämä suoritus vain kerran ohjelmaan. Joka kerta kun käsky GOSUB (engl. go to subroutine) suoritetaan, tietokone suorittaa käskyn ODOTA 3 s, jota sanotaan aliohjelmaksi. Aliohjelma, joka esiintyy riveillä 170—190, sisältää paitsi noin kolmen sekunnin viiveen, myös PRINT-lauseet, joilla selvennetään näyttöä. Rivi

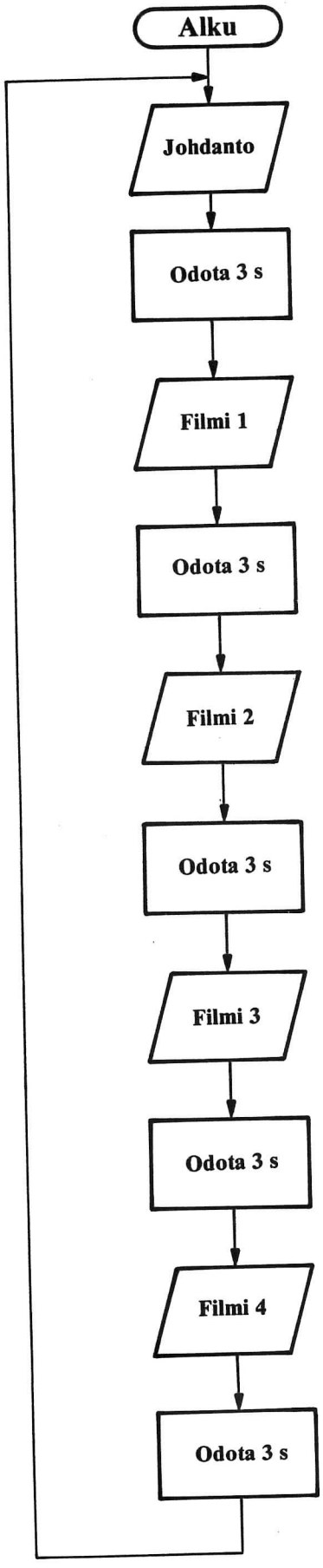
```
30 GOSUB 170
```

merkitsee sitä, että tietokoneen tulee suorittaa ohjelmanosa, joka alkaa riviltä 170. Jotta tietokone saisi tiedon siitä, missä aliohjelma päättyy, on sen loppuun kirjoitettava käsky RETURN, jolloin ohjelman suoritus palautuu sitä GOSUB-lausetta seuraavalle riville, jolta hyppy aliohjelmaan viimeksi on tapahtunut.

Ohjelmaksi saadaan

```
10 PRINT CHR$(12)
20 PRINT "ELOKUVAT TÄLLÄ VIIKOLLA:"
30 GOSUB 170
40 PRINT "PEPPI PITKÄTOSSU"
50 PRINT "SUNNUNTAINA 13-15,15-17"
60 GOSUB 170
70 PRINT "TULITIKUT HUKASSA"
80 PRINT "MAANANTAINA 19-21,21-23"
90 GOSUB 170
100 PRINT "ETELÄN TÄHTIEN ALLA"
110 PRINT "KESKIVIIKKONA 18-20,20-22"
120 GOSUB 170
130 PRINT "SEITSEMÄN KÄÄPIÖN PALUU"
140 PRINT "LAUANTAINA 13-15"
150 GOSUB 170
160 GOTO 10
170 PRINT "=============================="
180 PRINT
190 FOR V=1 TO 3000 : NEXT V : RETURN
```

Aja ohjelma!


Alku
Johdanto
Odota 3 s
Filmi 1
Odota 3 s
Filmi 2
Odota 3 s
Filmi 3
Odota 3 s
Filmi 4
Odota 3 s

Lisäyksiä ohjelmaan

Koska elokuvateatterin omistaja haluaa nyt ilmoittaa, että elokuvat n:o 2 ja 3 ovat lapsilta kiellettyjä, lisätään vielä yksi aliohjelma

```
200 PRINT "KIELLETTY LAPSILTA" : RETURN
```

Aliohjelmaa joudutaan kutsumaan kaksi kertaa ohjelmassa. Lisää esimerkiksi

```
35 GOSUB 200
115 GOSUB 200
```

Aja ohjelma ja tarkista, että tieto "KIELLETTY LAPSILTA" tulee näyttöön oikeissa paikoissa.

Omia kommentteja ohjelmassa

Jälkeenpäin voi olla vaikea ymmärtää pitkiä ohjelmia pelkän ohjelmalistan perusteella. Omia ohjelmiaan kirjoittava voi helpottaa työtään sijoittamalla ohjelmaan kommentteja käskyllä REM (engl. remark). Rivit

```
165 REM # ALIOHJELMA VIIVÄSTYS
195 REM ¤ ALIOHJELMA KIELLETTY LAPSILTA
```

voidaan siksi lisätä ohjelmaan. Näillä riveillä tietokone ei suorita mitään. Sitävastoin rivit ovat tietokoneen muistissa, ja ne saadaan normaalisti näyttöön LIST-käskyllä. Ohjelma on nyt kuitenkin niin suuri, että se ei mahdu yhtaikaa näyttöön. Käytä siis käskyä LIST-100 ohjelman alkuosan tarkastelemiseksi ja vastaavasti LIST 100-loppuosaa varten.

Aliohjelmat sijoitetaan pääohjelman ulkopuolelle

Sitä osaa ohjelmasta, joka ei sisällä lainkaan aliohjelmia, kutsumme pääohjelmaksi. Esimerkissämme pääohjelma käsittää rivit 10—160, kun aliohjelmat puolestaan ovat riveillä 170—200. Jotta ohjelma olisi havainnollisesti ymmärrettävissä, on tapana sijoittaa aliohjelmat aina viimeisiksi. Jos haluat, että ohjelma suoritetaan vain yhden kerran, on sinun kirjoitettava END-lause pääohjelman ja aliohjelmien väliin, esimerkiksi

```
160 END
```

Ohjelmassa, joka ei sisällä yhtään aliohjelmaa, ei tarvita END-lausetta. ABC 80 saa joka tapauksessa tiedon siitä, mikä on ohjelman viimeinen käsky.

Ohjelmarivien uudelleennumerointi

Alkuperäisessä ohjelmassa olivat kaikki rivinumerot täysiä kymmenlukuja: 10, 20, 30, jne. Jos haluat, että myös monien muutosten jälkeinen valmis ohjelma sisältää rivinumeroina vain täysiä kymmenlukuja, käytä käskyä REN (engl. renumber).

Kirjoita

```
REN
```

Tarkista käskyillä LIST-100 ja LIST 100-, mitä muutoksia rivinumeroissa on tapahtunut. Huomaa, että myös käskyjen GOSUB rivinumerot ovat muuttuneet niin, että jokaisen GOSUB käskyn perässä oleva numero on uuden tilanteen mukainen. Sama koskee myös käskyllä GOTO muodostettuja lauseita.

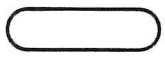
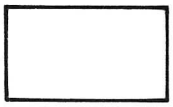
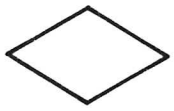
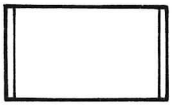
Näin sinun ei tarvitse erikseen huolehtia hyppyosoitteiden muutoksista uudelleennumeroinnin yhteydessä.

Yhteenveto

Käsky	Selitys	Esimerkki
GET	Lukee näppäimistöstä merkin jonomuuttujan arvoksi kirjoittamatta sitä näyttöön.	`90 GET A$`
GOSUB	Hyppää aliohjelmaan.	`30 GOSUB 90`
RETURN	Päättää aliohjelman ja aiheuttaa paluuhypyn pääohjelmaan riville, joka seuraa vastaavaa GOSUB-käskyä.	`120 RETURN`
END	Lopettaa ohjelmanajon.	`200 END`
REM	Ohjelmaan voidaan kirjoittaa kommentti.	`30 REM ** ALIOHJ 1`

Komento	Selitys	Esimerkki
REN	Numeroi uudestaan ohjelman kaikki rivit ja muuttaa vastaavasti käskyt GOTO, GOSUB jne.	
	Seuraavia vaihtoehtoja voidaan käyttää:	
	Numeroi rivit alkaen 10:stä ja jatkaen 10:n välein.	REN
	Numeroi rivit aloittaen tietystä numerosta, esim. 25, ja jatkaa 25:n välein.	REN 25
	Numeroi rivit aloittaen tietystä numerosta, esim. 500, ja jatkaa halutuin välein, esim. 20.	REN 500 , 20

Vuokaaviosymbolit

Symboli	Merkitys
	Ohjelman alku tai loppu. Vaihtoehto ilmoitetaan symbolissa.
	Rajoitettu toimintatehtävä, joka lyhyesti kuvaillaan symbolissa.
	Ohjelma jatkuu tietyllä rivillä sen mukaan, mikä on tässä tehtävän testin tulos. Testin ehto kirjoitetaan symboliin.
	Rajoitettu joukko tehtäviä, joita ei yksilöidä tässä kaaviossa. Ratkaisut ja tehtävät kuvataan erillisessä vuokaaviossa.
	Rajoitettu tehtävä, johon liittyy kuvaruutu, näppäimistö tai muu syöttö- tai tulostuslaite. Tehtävä ilmoitetaan symbolissa.
	Numeroitu ulosmeno vuokaaviosta. Vuokaavio jatkuu toisaalla, missä tämä symboli esiintyy samalla numerolla varustettuna.

Harjoituksia

7.1 Ohjelma

```
10 PRINT CHR$(7)
```

aiheuttaa äänimerkin, joka on jo tuttu virhetoimintojen yhteydestä. Tee ohjelma, jota voit käyttää esim. kananmunien keittoajan määrityksessä. Tietokone kysyköön halutun keittoajan ja ilmoittaa äänisignaalilla ajan kuluneen. Äänimerkki ei saa loppua ennen kuin keittäjä 'kuittaa' merkin näppäilemällä CTRL ja C samanaikaisesti. Tee myös ohjelman lohkokaavio.

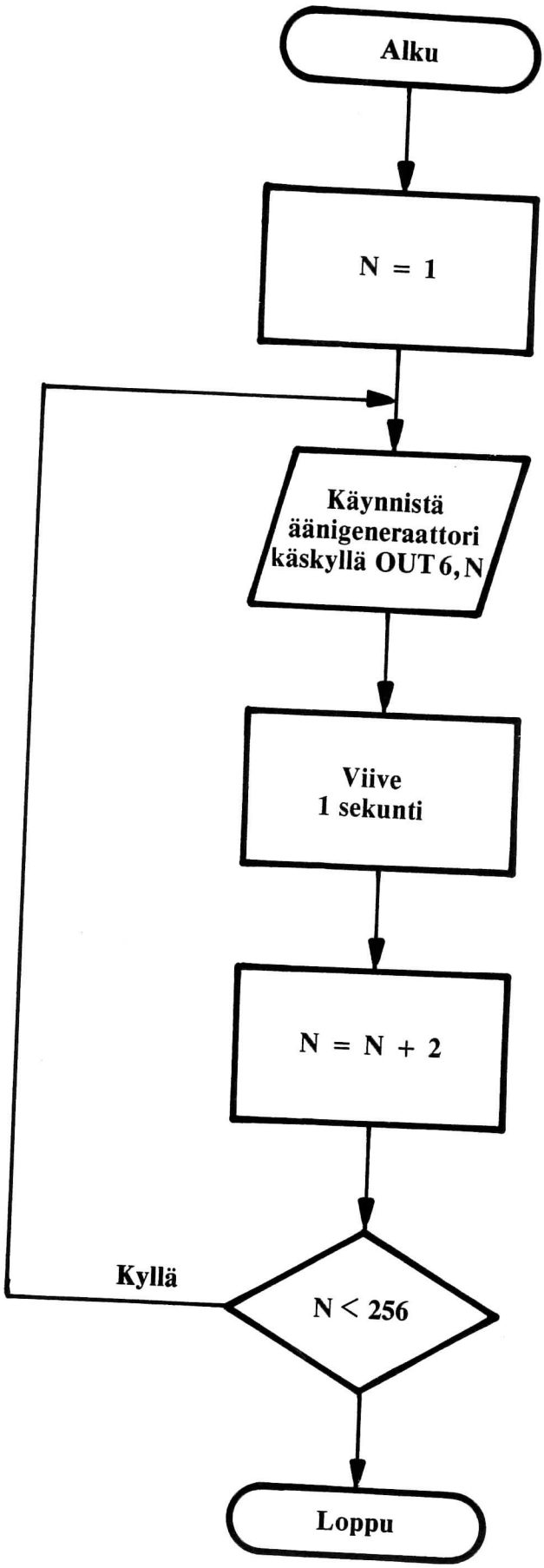
7.2 Käytä kappaleen 7 ensimmäistä ohjelmaa laskemaan, kauanko kestää auton, jonka hinta on nyt 50 000:—, nousta 1 miljoonaan, kun inflaatio on 9 %:

ABC-tietokoneessa on myös äänigeneraattori. Sen toiminnan saa aikaan BASIC-käsky OUT 6, N, missä N on pariton luku väliltä 1—255. Antamalla N:lle eri arvoja saat aikaan erilaisia äänitehosteita. Käsky OUT 6,0 sulkee generaattorin. Seuraava pieni ohjelma esittelee äänigeneraattorin toimintaa, kun N = 7:

```
10 OUT 6,7 : REM **ALOITA ÄÄNIMERKKI**
20 FOR V=0 TO 5000 : NEXT V
30 OUT 6,0 : REM **LOPETA ÄÄNIMERKKI**
```

Piirrä ylläolevan ohjelman vuokaavio. Kokeile ohjelmaa myös muilla N:n arvoilla.

7.4 Kirjoita ohjelma seuraavan sivun lohkokaavion mukaisesti.


Alku
N = 1
Käynnistä
äänigeneraattori
käskyllä OUT 6, N
Viive
1 sekunti
N = N + 2
Kyllä
N < 256
Loppu

7.5 Seuraava ohjelma saa aikaan kolme eri ääntä, kun painat näppäimiä A, B ja C kutakin vuorollaan:

```
10 GET A$
20 IF A$="A" THEN GOSUB 100
30 IF A$="B" THEN GOSUB 200
40 IF A$="C" THEN GOSUB 300
50 GOTO 10
100 OUT 6,5 : RETURN
200 OUT 6,3 : RETURN
300 OUT 6,251 : RETURN
```

Täydennä ohjelmaa siten, että myös näppäinten D, E ja F painallus saa aikaan eli generoi uuden, niihin liitetyn äänitehosteen.

KAPPALE 8

Tietokone käsittelee kirjainyhdistelmiä

ABC80

Tietokone kirjoittaa kirjeitä

Hyvin tavallinen tietokoneiden käyttöalue on panna ne kirjoittamaan tuhansia "henkilökohtaisia" kirjeitä mm. mainosmielessä. Tarkastelkaamme ohjelmaa, jossa tietokonetta käytetään hyväksi kirjoitettaessa tietynmuotoisia kirjeitä, joihin saadaan henkilökohtainen luonne siten, että sopiviin paikkoihin liitetään nimeä vastaava merkkijonoteksti. Kirjeitä kirjoitettaessa voidaan käyttää hyväksi pienten kirjainten käyttömahdollisuutta. Siispä paina näppäintä [UPPER CASE] niin, että valodiodi sammuu. Näppäimistö toimii nyt kuin tavallisessa kirjoituskoneessa, joten aakkosnäppäimillä tuotetaan normaaleja pieniä kirjaimia. Tarvittaessa saadaan isoja kirjaimia näppäimen SHIFT avulla. Jos haluat, voit toki kirjoittaa BASIC-kieliset käskyt ja komennot myös pienillä kirjaimilla, sillä ABC 80 automaattisesti muuttaa BASIC-kielen käskyt isoiksi kirjaimiksi. Kokeile! Sitten kun et enää tarvitse pieniä kirjaimia, palauta näppäimistö tilaan "UPPER CASE".

```
10 PRINT "KENELLE KUTSU LÄHETETÄÄN";
20 INPUT A$
30 PRINT CHR$(12)
40 PRINT "Hei ";A$
50 PRINT
60 PRINT "Olet ";A$;" tervetullut"
70 PRINT "syntymäpäiväkutsuilleni"
80 PRINT "lauantaina klo 14.00 ."
90 PRINT "Nähdään ";A$;" silloin !"
100 PRINT
110 PRINT "ystävällisin terveisin Maija"
```

Vaihtele PRINT-lauseita, niin saat aikaan haluamasi kirjeen.

Jonoalkio (jonomerkki)

Lainausmerkeillä eli sitaateilla ympäröityä merkkien jonoa kutsutaan lyhyesti merkkijonoksi. Ilman lisätilan varaamista eli dimensiointia voidaan kaikkia enintään 80 merkin merkkijonoja käsitellä merkkijonomuuttujan avulla. Tilanvarauskäskyn DIM avulla voivat merkkijonot olla mielivaltaisen pitkiä. Jos haluamme käsitellä merkkijonossa olevia yksityisiä eri merkkejä, on meidän tunnettava seuraava periaate, jolla jonon merkit numeroidaan.

```
Merkin
paikkanumero  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 2
Jono          "B A S I C - o h j e l m o i n n i n   A B C - k i r j a
```

Ylläolevassa esimerkissä "BASIC-ohjelmoinnin ABC-kirja" on merkki B n:o 1 ja merkki a n:o 28.

Jonofunktiot LEFT¤, RIGHT¤ ja MID¤

Kun nyt tiedämme, kuinka jonon eri merkit ilmoitetaan ja paikallistetaan, voimme alkaa käyttää erityisiä merkkijonofunktioita, joita ABC 80 -tietokoneessa on.

Sijoitamme jonomuuttujalle A¤ arvon seuraavalla tavalla:

```
10 A¤="BASIC-ohjelmoinnin ABC-kirja"
```

Ensimmäisenä jonofunktiona käytämme seuraavassa funktiota LEFT¤(A¤, J). Tämä funktio antaa tuloksena J ensimmäistä merkkiä jonosta A¤.

Rivi

```
20 B¤=LEFT¤(A¤,6)
```

sijoittaa jonomuuttujalle B¤ arvon "BASIC-"

Seuraava jonofunktio on RIGHT¤(A¤, J), joka antaa jonosta A¤ kaikki merkit J:nnestä merkistä alkaen. Rivi

```
30 C¤=RIGHT¤(A¤,20)
```

sijoittaa jonomuuttujalle C¤ arvon "ABC-kirja".

Voidaksemme sijoittaa jonomuuttujalle D¤ arvon "ohjelmoinnin" käytämme jonofunktiota MID¤(A¤, I, L), joka antaa L merkkiä jonon A¤ I:nnestä merkistä alkaen (se mukaanluettuna).

```
40 D¤=MID¤(A¤,7,12)
```

Päätä ohjelma sopivilla PRINT-lauseilla niin, että saadaan seuraava näyttö:

```
BASIC-
OHJELMOINNIN
ABC-KIRJA
```

Kuinka ohjelmaa pitää muuttaa, jotta näyttöön saadaan seuraava muoto:

```
BASIC-OHJELMOINNIN
ABC-KIRJA
```

Toteat, että voimme käsitellä merkkijonon sisältöä eri tavoilla käyttämällä esitettyjä jonofunktioita. Mietipä, kuinka seuraava ohjelma muuttaa jonon A¤ tulostusta.

```
10 A¤="BASIC-OHJELMOINNIN ABC-KIRJA"
20 FOR I=1 TO 28
30 IF MID¤(A¤,I,1)="-" THEN PRINT "--";
40 PRINT MID¤(A¤,I,1);
50 NEXT I
```

Kun ajat ohjelman, huomaat, että saamme nyt kolme väliviivaa yhden asemesta kaikkien yhdyssanojen väliin. Se johtuu siitä, että testaamme rivillä 30 jonon A¤ jokaisen merkin. Kun tietokone toteaa, että merkkinä on väliviiva, se kirjoittaa kaksi ylimääräistä väliviivaa näyttöön.

Sovellutukset, joissa tietokone kirjoittaa kirjeitä ja käsittelee kirjainyhdistelmiä, ovat nykyään hyvin tavallisia. On monia räätälintyönä tehtyjä järjestelmiä pelkästään sanojen käsittelemiseksi, mutta monet yleiskäyttöön tarkoitetut tietokoneet — kuten ABC 80 — soveltuvat mainiosti myös näihin tehtäviin, kunhan käyttäjä vain tekee tarkoitukseensa soveltuvat ohjelmat tai ostaa ne valmiina.

Kirjaimilla on arvo

Tietokone voi sisäisesti käsitellä ainoastaan numeroita. Tämän vuoksi jokaista kirjainta ja muuta merkkiä vastaa tietokoneen sisällä tietty numeroyhdistelmä. Yksi menetelmä numeroida kirjaimet ja merkit on käyttää hyväksi ASCII-koodia (American Standard Code for Information Interchange). ASCII luetaan usein "askii". Seuraavalla aukeamalla annetaan ASCII-koodit niille merkeille, joita ABC 80 -koneessa käytetään.

Monet merkkijonofunktiot käyttävät hyväksi sitä ominaisuutta, että jokaista merkkiä vastaa tietty numeroyhdistelmä.

Merkkijonofunktio ASC(A¤) antaa jonon A¤ ensimmäiselle merkille sen ASCII-arvon. Aja seuraava ohjelma ja tarkista tulos ASCII-taulukosta.

```
10 INPUT A¤
20 PRINT ASC(A¤)
30 IF A¤="LOPPU" THEN 50
40 GOTO 10
50 END
```

Halutessasi lopettaa ajon syötä A¤:lle arvoksi "LOPPU".

Merkkijonofunktio CHR¤ (I) antaa tulokseksi sen merkin, jonka ASCII-koodi on I.

Rivi

```
10 PRINT CHR¤(43)
```

tulostaa näyttöön "+" -merkin. Jonofunktio CHR¤ voi kerrallaan sisältää enintään neljä ASCII-koodia.

Minkä tulostuksen antaa seuraava rivi?

```
10 PRINT CHR¤(76,111,109,97)
```

Tarkastelkaamme vielä yhtenä sovelluksena, kuinka tietokone saadaan kirjoittamaan tietty lukumäärä umpimähkäisiä kolmekirjaimisia sanoja.

```
10 REM TEKSTIN GENEROINTI
20 RANDOMIZE
30 FOR X=1 TO 60
40 FOR I=1 TO 3
50 A(I)=INT(65+29*RND)
60 NEXT I
70 PRINT CHR¤(A(1),A(2),A(3)),
80 NEXT X
```

Rivinumero	**Rivin toiminto**
10	Kommentti.
20	Antaa funktiolle RND umpimähkäisen alkuarvon.
30, 40, 60, 80	Saavat aikaan kaksi silmukkaa.
50	Sijoittaa kenttään eli vektoriin A satunnaisluvun väliltä 65—93. Mieti tarkkaan, miksi!
70	Kirjoittaa näyttöön kolme isoa kirjainta. ASCII-taulukosta voidaan tarkistaa, että kaikki isot kirjaimet vastaavat lukuväliä 65—93.

ASCII-taulukko

ASCII	Merkki	ASCII	Merkki	ASCII	Merkki
7	Äänimerkki	59	;	93	Å
10	Rivinsiirto (LF)	60	<	94	Ü
		61	=	95	_
12	Kuvaruudun tyhjennys (FF)	62	>	96	é
		63	?	97	a
13	Vaunun palautus (CR)	64	É	98	b
		65	A	99	c
32	Välilyönti (SPACE)	66	B	100	d
33	!	67	C	101	e
34	"	68	D	102	f
35	#	69	E	103	g
36	¤	70	F	104	h
37	%	71	G	105	i
38	&	72	H	106	j
39	'	73	I	107	k
40	(	74	J	108	l
41	)	75	K	109	m
42	*	76	L	110	n
43	+	77	M	111	o
44	,	78	N	112	p
45	-	79	O	113	q
46	.	80	P	114	r
47	/	81	Q	115	s
48	0	82	R	116	t
49	1	83	S	117	u
50	2	84	T	118	v
51	3	85	U	119	w
52	4	86	V	120	x
53	5	87	W	121	y
54	6	88	X	122	z
55	7	89	Y	123	ä
56	8	90	Z	124	ö
57	9	91	Ä	125	å
58	:	92	Ö	126	ü
				127	■

Tietokone vertailee merkkijonoja

Kun olet nyt havainnut, että tietokone osaa tulkita kunkin kirjaimen vastaavana ASCII-koodin arvona, on helpompaa ymmärtää, että voimme myös vertailla jonoja keskenään sekä esimerkiksi järjestää niitä aakkosjärjestykseen. Seuraavat säännöt ovat voimassa:

Kun verrataan keskenään kahta yhtä pitkää jonoa, verrataan aluksi keskenään kummankin ensimmäisen merkin ASCII-koodia. Se ratkaisee, kumpi jonoista on "suurempi". Jos ensimmäiset merkit ovat samoja, verrataan keskenään toisia merkkejä samalla tavalla jne.

Kun verrataan keskenään kahta eripituista jonoa, suoritetaan edellä esitetty vertailu vain lyhyemmän jonon merkkimäärän mukaan. Jos lyhyemmän jonon mukaiset merkit ovat täysin samoja, määritellään pitempi jono "suuremmaksi".

Kaikkia aikaisemmin käytettyjä vertailurelaatioita voidaan käyttää myös jonojen välillä.

Jos esimerkiksi A¤ = "AB", B¤ = "ABC" ja C¤ = "BC",

niin on voimassa
A¤ < C¤
A¤ < B¤
B¤ < C¤

Nimien lajitteleminen

Monien nimien aakkostaminen käsin voi olla hyvin vaivalloista. Käyttämällä hyväksi tietokoneen nopeaa kykyä vertailla merkkijonoja voimme antaa tehtävän sen huoleksi. Aakkostamisessa voimme käyttää jo aikaisemmin esittämäämme ohjelmarakennetta. Syöttöä voimme kuitenkin yksinkertaistaa käyttämällä käskyä INPUTLINE. Kun jonomuuttujalle sijoitetaan arvo käskyllä INPUTLINE, niin koko rivi välilyönteineen ja muine merkkeineen tulee syötetyksi eikä tekstiä tarvitse sijoittaa lainausmerkkeihin. Myös rivinsiirtoa ja vaunupalautusta ilmaisevat merkit sisältyvät jonoon, mikä merkitsee sitä, että myös ylimääräiset rivin siirrot saadaan näyttöön.

```
10 PRINT "MONTAKO NIMEÄ LAJITELLAAN";
20 INPUT N
30 FOR X=1 TO N
40 INPUTLINE A¤(X)
50 PRINT
60 NEXT X
70 FOR I=1 TO N-1
80 FOR J=I+1 TO N
90 IF A¤(I)<A¤(J) THEN 110
100 X¤=A¤(I) : A¤(I)=A¤(J) : A¤(J)=X¤
110 NEXT J
120 NEXT I
130 PRINT CHR¤(12)
140 FOR X=1 TO N
150 PRINT A¤(X)
160 NEXT X
```

Syötä aluksi koneelle tieto siitä, kuinka monta nimeä sen tulee aakkostaa ja sen jälkeen kyseeseen tulevat nimet. Viimeisen nimen syöttämisen jälkeen kone tulostaa välittömästi aakkosissa olevan luettelon. Jos haluat syöttää enemmän kuin 10 nimeä, on sinun lisättävä ohjelmarivi, joka laajentaa muuttujan A¤ kentän.

```
5 DIM A¤(20)=30
```

Tämän rivin lisäämisen jälkeen voidaan syöttää 20 nimeä järjestettäviksi. Samalla määräämme, että kukin nimi saa sisältää enintään 30 merkkiä. Tämän rajoituksen avulla vältämme tarpeettoman muistitilan varaamisen tietokoneen muistista.

Onko edellä olevassa ohjelmassa mitään virhevaaraa? Tarkistapa kirjaimien Å, Ä ja Ö järjestys ASCII-taulukossa! Jotta mainitut kirjaimet saataisiin aakkostumaan oikein, täytyy ohjelmaaan tehdä lisäys. Yksi ehdotus annetaan tämän luvun harjoitustehtävässä 8.5.

Lajittelu pituuden mukaan

Jonofunktio LEN(A¤) antaa merkkijonon A¤ pituuden. (engl. length, pituus).

```
10 A¤="ABC-KIRJA"
20 PRINT LEN(A¤)
```

Ylläoleva ohjelma tulostaa näyttöön luvun 9, joka ilmoittaa jonossa A¤ olevien kaikkien merkkien todellisen lukumäärän. Jonofunktiota LEN(A¤) voidaan

käyttää lajittelemaan merkkijonoja kasvavan pituuden mukaiseen järjestykseen. Olettakaamme esimerkiksi, että liikkeenharjoittaja haluaa esittää "päivän erikoistarjoukset" ABC 80 -tietokoneensa avulla. Mietipä, kuinka edellisen kappaleen lajitteluohjelma tulisi muunnella, jotta syötetyt merkkijonot tulostettaisiin kasvavan pituuden mukaisessa järjestyksessä!

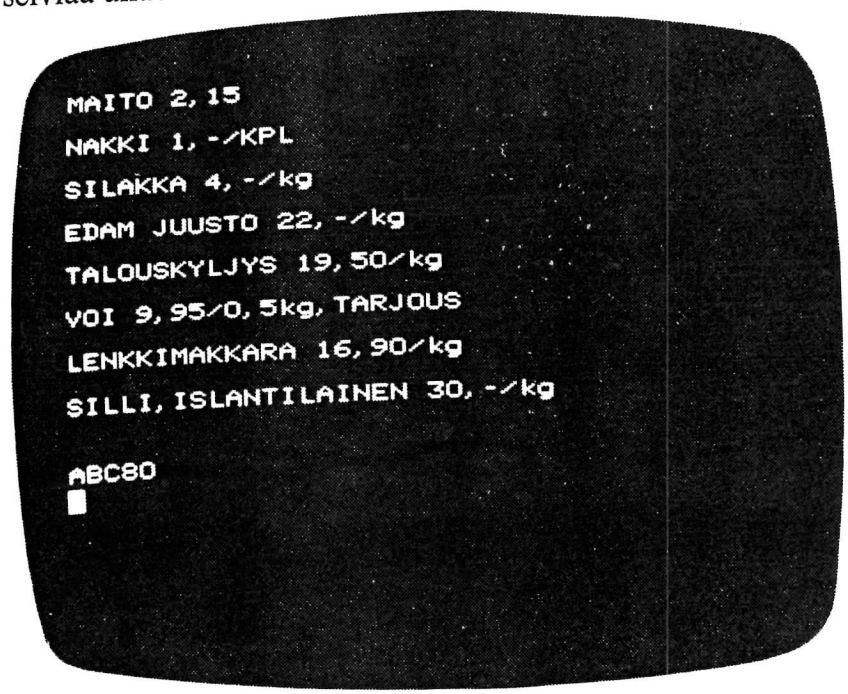
Ehdotus muunnetuksi ohjelmaksi:

```
5 DIM A¤(20)=30
10 PRINT "KUINKA MONTA ARTIKKELIA";
20 INPUT N
30 FOR X=1 TO N
40 INPUTLINE A¤(X)
50 PRINT
60 NEXT X
70 FOR I=1 TO N-1
80 FOR J=I+1 TO N
90 IF LEN(A¤(I))<LEN(A¤(J)) THEN 110
100 X¤=A¤(I) : A¤(I)=A¤(J) : A¤(J)=X¤
110 NEXT J
120 NEXT I
130 PRINT CHR¤(12)
140 FOR X=1 TO N
150 PRINT A¤(X)
160 NEXT X
```

Kun edellä olevaa ohjelmaa käytettiin lihakauppiaan hyödyksi, käytettiin seuraavaa kahdeksaa syöttöarvoa:

```
KUINKA MONTA ARTIKKELIA? 8
NAKKI 1,-/KPL
TALOUSKYLJYS 19,50/kg
LENKKIMAKKARA 16,90/kg
EDAM JUUSTO 22,-/kg
SILAKKA 4,-/kg
SILLI,ISLANTILAINEN 30,-/kg
VOI 9,95/0,5 kg,TARJOUS
MAITO 2,15
```

Ajon tulos selviää allaolevasta kuvasta.



Jonon pituuden mukainen järjestys

Merkkijonojen liittäminen yhteen

Ainoa matemaattinen operaatio, joka voidaan suorittaa merkkijonoilla, on yhteenlasku. Seuraava esimerkki kuvaa tätä toimintoa

```
10 PRINT "HEI"+"HEI"
```

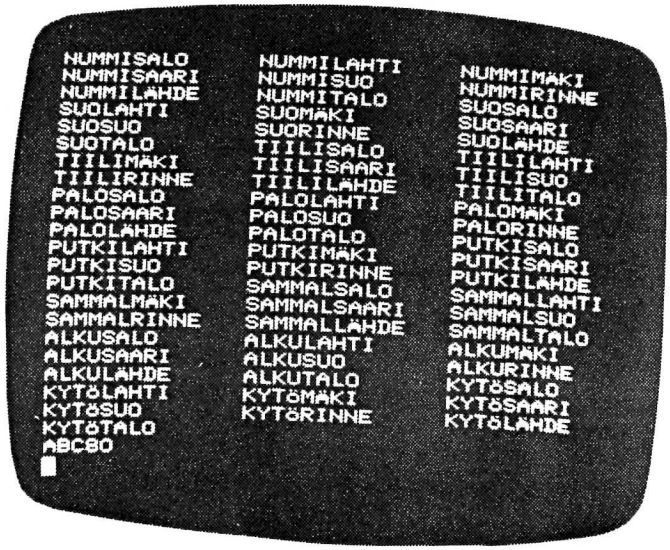
Seuraava esimerkki kuvaa erinomaisesti, kuinka merkkijonojen yhdistämistä voitaisiin käytännöllisesti hyödyntää vaikkapa tulevassa (?) Valtion Nimiasiain Keskusvirastossa (VNK) etsittäessä uuden nimiasetuksen mukaisia avioparin itsensä valitsemiin perinteisiin ja luonnonläheisiin sanoihin perustuvia sukunimiä

```
10 FOR I=1 TO 8
20 INPUT A¤(I),B¤(I)
30 NEXT I
40 FOR I=1 TO 8
50 FOR J=1 TO 8
60 PRINT A¤(I)+B¤(J),
70 NEXT J
80 NEXT I
```

Ohjelman riveillä 10, 20 ja 30 muodostuu silmukka, jossa syötetään kahdeksan merkkijonoa kenttään A¤ nimen alkuosaksi ja kahdeksan merkkijonoa kenttään B¤ nimiehdoksen luontevaksi loppuosaksi. Koeaja ohjelma esimerkiksi seuraavilla sanapareilla.

```
NUMMI,SALO
SUO,LAHTI
TIILI,MÄKI
PALO,SAARI
PUTKI,SUO
SAMMAL,RINNE
ALKU,LÄHDE
KYTÖ,TALO
```

Ajon tuloksena kuvaruudussa on 64 yhdysnimikombinaatiota, joista onnellisen avioparin on helppo valita se ainoa oikea sukunimi.



Jonoista tulee lukuja

Joskus on tarpeellista kyetä suorittamaan laskutoimituksia numeroilla, jotka sisältyvät merkkeinä merkkijonoihin. Tätä varten on käytettävissä funktio VAL (A¤), joka muuttaa merkkijonossa olevan numeron luvuksi (engl. value, arvo).

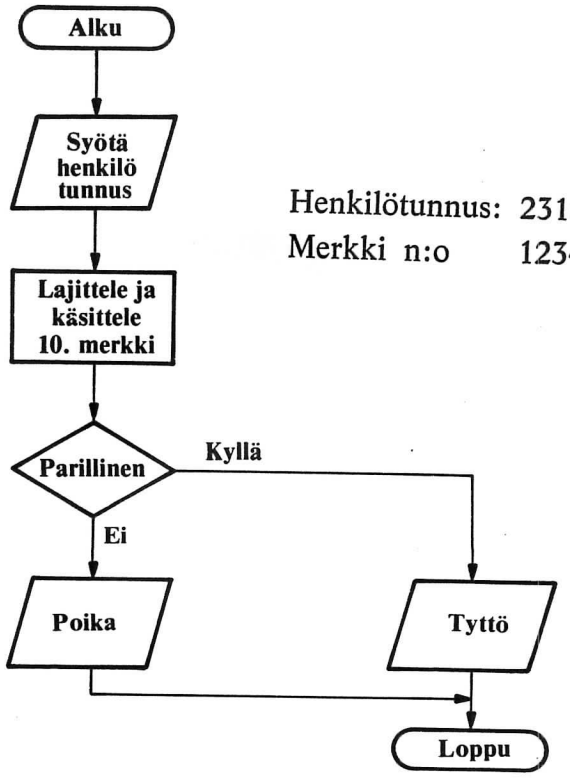
```
10 A¤="5"
20 Z=VAL(A¤)
30 PRINT Z*Z*2
```

Ylläolevassa ohjelmassa sijoitetaan jonomuuttujalle A¤ sisällöksi numeromerkki 5. Tämän jälkeen annetaan liukulukumuuttujalle Z arvo 5 jonofunktion VAL (A¤) avulla. Liukulukumuuttujaa voidaan käyttää aritmeettisessa lausekkeessa, ja niinpä ohjelmanajo antaa näyttöön 50.

Allaoleva esimerkki jonofunktion VAL (A$) käytöstä on huomattavasti käytännönläheisempi. Kuten ehkä tiedät, voi henkilötunnuksestasi heti todeta, oletko mies- vai naispuolinen henkilö. Monissa virallisissa kyselykaavakkeissa ei vain vielä ole osattu hyödyntää tätä, vaan tieto ilmoitetaan merkitsemällä risti oikeaan ruutuun.
Kokeile seuraavaa ohjelmaa ja totea itse, onko sukupuolesi määritelty oikein.

```
10 PRINT "SYÖTÄ HENKILÖTUNNUKSESI"
20 PRINT
30 INPUT A$
40 B$=MID$(A$,10,1)
50 B=VAL(B$)
60 IF B/2=INT(B/2) THEN 100
70 PRINT
80 PRINT "OLET POIKA"
90 GOTO 120
100 PRINT
110 PRINT "OLET TYTTÖ"
120 PRINT
```

Ohjelma testaa annetun henkilötunnuksen toiseksi viimeisen eli 10. merkin arvon käyttämällä VAL-jonofunktiota. Jos se on pariton, tietokone tulostaa "OLET POIKA", jos taas se on parillinen, tulostus on "OLET TYTTÖ". Ohjelman toimintaa kuvaa seuraava lohkokaavio.



Henkilötunnus: 231045-569V
Merkki n:o 12345678901

Luvuista tulee jonoja

Jonofunktio NUM¤(A) muuttaa luvut merkkijonoiksi. Näin ollen tämä funktio on toiminnaltaan käänteinen funktiolle VAL(A¤). Käskyllä NUM¤(A) saadussa jonossa on ne merkit, jotka olisi saatu tulostamalla muuttuja A käskyllä PRINT.

```
10 A=1234
20 A¤=NUM¤(A)
30 FOR X=1 TO 5
40 PRINT MID¤(A¤,X,1);" ";
50 NEXT X
```

Kun näyttöön kirjoitetaan positiivisia lukuja PRINT-lauseilla, tulee näyttöön välilyönti plus-merkin sijasta. Siksi tämä välilyönti sisältyy jonoon NUM¤(A). Edellä olevassa esimerkissä saadaan siis NUM¤(A) = " 1234", eli jono sisältää viisi alkiota.

Muuttamalla liukulukumuuttuja A jonomuuttujaksi A¤ voimme vaikuttaa numeroyhdistelmien tulostukseen aikaisemmin käyttämillämme jonofunktioilla. Tässä tapauksessa numerot kirjoitetaan näyttöön kahdella välilyönnillä jokaisessa välissä. Voimme myös helpommin testata, sisältyykö jokin numero annettuun lukuun muuttamalla se merkkijonoksi.

Automaattinen alleviivaus

Jonofunktio STRING¤(L, A) antaa merkkijonon, joka sisältää L kappaletta arvoa A vastaavaa ASCII-merkkiä. Esimerkiksi käsky STRING¤(5,42) tuottaa jonon " ***** ". Jonofunktiota STRING¤(L, A) voidaan käyttää esimerkiksi jonojen automaattiseen alleviivaukseen, kuten on tehty seuraavassa ohjelmassa.

```
10 INPUT A¤
20 PRINT CHR¤(12)
30 PRINT A¤
40 L=LEN(A¤)
50 PRINT STRING¤(L,61)
```

Mitä merkkiä on tässä käytetty alleviivaukseen? Vaihtelepa riviä 50 niin, että "alleviivaukseen" käytetään muita merkkejä.

Tietokone hakee sanoja

Jonofunktion INSTR(I,A¤,B¤) avulla on helppo tutkia, sisältyykö tietty merkmerkkiyhdistelmä johonkin tutkittavaan jonoon.

INSTR(I,A¤,B¤) etsii jonoa B¤ jonosta A¤ aloittaen merkistä numero I. Funktion toiminnon tulokseksi tulee se jonon A¤ paikka, jossa B¤ ensimmäisen kerran esiintyy, tai 0, jos B¤ ei lainkaan sisälly jonoon A¤.

Jos A¤ = "ABABCAB" ja B¤ = "AB", antaa funktio INSTR(2,A¤,B¤) tuloksen 3, sillä vertailuhan alkoi merkistä n:o 2 alkaen.

Ennen kuin tutkimme yksinkertaista esimerkkiä, määrittelemme jonofunktion SPACE¤ (I) (engl. space). Se tuottaa peräkkäin I kappaletta välilyöntejä.

Seuraavassa esimerkkiohjelmassa testataan, sisältyykö teksti BASIC syötettyyn merkkijonoon .Jos näin on, tulostetaan jono, ja siinä oleva teksti BASIC alleviivataan. Muussa tapauksessa ilmoitetaan, että teksti BASIC ei sisälly jonoon.

```
10 INPUT A¤
20 B¤="BASIC"
30 X=INSTR(1,A¤,B¤)
40 IF X=0 THEN 90
50 PRINT CHR¤(12)
60 PRINT A¤
70 PRINT SPACE¤(X-1);"====="
80 GOTO 100
90 PRINT "SANA 'BASIC' EI SISÄLLY
LAUSEESEEN"
100 PRINT
```

Yhteenveto

Käsky	Selitys	Esimerkki
INPUTLINE	Lukee näppäimistöllä syötetyn rivin täydellisenä välilyönteineen ja lainausmerkkeineen. Myös vaunupalautusta ja rivin tulostusta koskevat merkit sisältyvät loppuun.	`50 INPUTLINE A¤`
DIM	Varaa tilaa muistista. Eri vaihtoehtoja:	
	Jonon pituuden määrittely.	`10 DIM A¤=200`
	Muuttujan kentän (vektorin) määrittely.	`30 DIM A(20)` `40 DIM A%(50)`
	Jonon kentän määrittely.	`50 DIM A¤(20)=30`

Jonofunktio	Selitys	Esimerkki
LEFT¤(A¤, J)	Antaa J ensimmäistä merkkiä jonosta A¤.	10 C¤=LEFT¤(A¤, J)
MID¤(A¤, I, J)	Antaa J merkkiä I:nnestä merkistä alkaen.	20 D¤=MID¤(A¤, I,J)
RIGHT¤(A¤, I)	Antaa kaikki jonon A¤ merkit alkaen I:nnestä merkistä.	30 D¤=RIGHT¤ (A¤,I)
LEN(A¤)	Antaa jonon A¤ pituuden.	40 L=LEN(A¤)
ASC(A¤)	Antaa jonon A¤ ensimmäisen merkin ASCII-arvon.	50 V=ASC(A¤)
CHR¤(I)	Antaa merkin, jonka ASCII-arvo on I. Voi sisältää enintään neljä merkkiä.	10 PRINT CHR¤(7) 5 PRINT CHR¤(65, 66,67)
INSTR(I, A¤,B¤)	Etsii jonoa B¤ jonosta A¤ alkaen A¤:n I:nnestä merkistä. Funktio ilmoittaa paikan, jossa B¤ esiintyy ensimmäisen kerran tai antaa arvon 0, jos B¤ ei esiinny A¤:ssa.	60 P=INSTR(I,A¤, B¤)
SPACE¤(I)	Antaa peräkkäin I kpl välilyöntejä.	50 PRINT SPACE¤ (I)
STRING¤(I, C)	Antaa jonon, jossa on I kpl merkkejä CHR¤(C).	60 PRINT STRING¤ (I,C)
NUM¤(A)	Antaa merkkijonon, joka saataisiin kirjoittamalla näyttöön A käskyllä PRINT.	70 A¤=NUM¤(A)
VAL(A¤)	Antaa merkin A¤ arvon luvuksi tulkittuna.	80 A=VAL(A¤)
+	Liittää jonoja yhteen.	10 A¤=B¤+C¤+"!"

Harjoituksia

8.1 Tee ohjelma, joka kysyy

```
MIKÄ SINUN NIMESI ON ?
```

Tämän jälkeen ohjelman on INPUT-käskyllä ja muuttujan A¤ avulla tulostettava syötetty merkkijono — esim. TARJA — seuraavan lisätiedon kanssa:

```
HEI TARJA
TARJA ON KIVA NIMI
```

8.2 Kirjoita ohjelma, jossa muuttuja A¤ saa arvon "OHJELMOINNIN ABC-KIRJA". Käytä sen jälkeen sopivia merkkijonofunktioita siten, että ohjelma tulostaa

```
OHJELMOINNIN - ABC - KIRJA
```

8.3 Tee ohjelma, jossa INPUT-lauseella syötetään muuttujan A¤ arvoksi henkilötunnus eli ns. sosiaaliturvatunnus. Poimi sen jälkeen tunnuksesta syntymävuotta osoittavat luvut ja tulosta

```
OLET SYNTYNYT VUONNA 1945
```

8.4 Tee ohjelma, joka tulostaa 60 V-alkuista kirjainyhdistelmää, joissa kussakin on kolme kirjainta. Ohjelmaa voitaisiin luultavasti hyödyntää keksittäessä valtion virastoille sopivia lyhenteitä.

8.5 Alla oleva ohjelma lajittelee syötetyt nimet oikeaan järjestykseen, myös kirjaimet Å, Ä ja Ö. Tiettyjä rajoituksia lajittelussa on kuitenkin edelleen. Aja ohjelma eri nimillä ja yritä todeta esiintyvät rajoitukset.

```
10 PRINT "MONTAKO NIMEÄ LAJITELLAAN";
20 INPUT N
30 FOR X=1 TO N
40 INPUTLINE A$(X)
50 PRINT
60 NEXT X
70 FOR I=1 TO N-1
80 FOR J=I+1 TO N
90 IF A$(I)<A$(J) THEN 110
100 X$=A$(I) : A$(I)=A$(J) : A$(J)=X$
110 NEXT J
120 NEXT I
130 PRINT CHR$(12)
140 FOR X=1 TO N
150 IF ASC(A$(X))=91 THEN 180
160 IF ASC(A$(X))=92 THEN 180
170 PRINT A$(X)
180 NEXT X
190 FOR X=1 TO N
200 IF ASC(A$(X))=91 THEN PRINT A$(X)
210 IF ASC(A$(X))=92 THEN PRINT A$(X)
220 NEXT X
```

8.6 Kirjoita ohjelma, joka lukee sisään merkkijonon — esim. nimen — ja tulostaa sen seuraavasti kehystettynä:

```
**********
*OTSIKKO*
**********
```

8.7 Eräässä armeijan juhlatilaisuudessa päätettiin kaikki osallistuvat vääpelit ylentää ylivääpeleiksi. Kirjoita ohjelma, joka lukee sisään allaolevan osanottajalistan arvonimineen:

VÄÄPELI RISTO RÄMÄ
VÄÄPELI NIILO MERI
VÄÄPELI MATTI MÄKI
VÄÄPELI VILLE VALO

Ohjelman on sen jälkeen tulostettava sama lista siten, että kaikki osanottajat on nimitetty ylivääpeleiksi.

8.8 Paljon julkisuudessa esiintyvällä johtajalla oli vaikeuksia lausua oikein S-äänne. Siksi hän oli määrännyt sihteerinsä tarkastamaan kaikki hänen kirjoitetut puheensa ja valvomaan, ettei S-kirjainten lukumäärä millään rivillä ole yli viisi prosenttia kaikista rivin kirjaimista. Tee sihteerin avuksi ohjelma, joka testaa kirjoitetun rivin S-kirjainten esiintymisen sekä pyytää kirjoittamaan rivin uudelleen, mikäli se on yli 5 %.

8.9 Eräässä pohjoiseurooppalaisessa kuningaskunnassa tapahtui vallanvaihto, jolloin vanha kuningas Kustaa Vanha luopui kruunustaan. Uusi kuningas Kalle Kustaa määräsi, että kaikissa asiakirjoissa ja kirjeissä pitää vanha merkintä KUSTAA VANHA korvata uuden hallitsijan nimellä KALLE KUSTAA.
Kirjoita ohjelma, joka enintään 120 merkin mittaisesta kirjoitetusta lauseesta muuttaa mahdollisesti esiintyvän nimen KUSTAA VANHA uudeksi KALLE KUSTAA'ksi. Nimi saa esiintyä vain kerran kussakin kirjoitetussa lauseessa.

KAPPALE 9

Tietokone ratkaisee matemaattisia ongelmia

ABC80

Lisää laskentamahdollisuuksia

ABC 80 ei osaa pelkästään neljää peruslaskutoimitusta, se osaa myös ilmoittaa suoraan tavallisimpien matemaattisten funktioiden arvot. Ne funktiot, jotka ABC 80 pystyy määrittämään, ovat potenssifunktiot, trigonometriset funktiot, logaritmifunktiot, eksponenttifunktiot sekä neliöjuuri. Lisäksi ABC 80:ssa voi käyttää joitain erityisfunktioita kuten kokonaisosa, itseisarvo, satunnaisluku ym.

Kaikkia funktioita voidaan käyttää täsmälleen samalla tavalla kuin lukuja ja muuttujia käytetään BASIC-lauseissa, ovathan ne itseasiassa vain jostakin annetusta lukumuuttujasta riippuvia lukuarvoja.

Potenssiin korotus

Olet oppinut 4. luvussa, kuinka luvun 10 potensseja käytetään luvun esittämiseksi. ABC 80 -tietokonetta voidaan myös käyttää mielivaltaisen potenssin laskemiseen. Ottakaamme esimerkiksi 2^8, joka on potenssimerkintä. Lukua 2 sanomme kantaluvuksi ja lukua 8 eksponentiksi.

ABC 80 ymmärtää potenssimerkinnälle kaksi erilaista BASIC-kielen merkintää. Nämä symbolit ovat "Ü" (huomaa, että symbolina on iso Ü kirjain) ja "**".

Esimerkiksi 2^8
kirjoitetaan 2 Ü 8 %
tai vaihtoehtoisesti 2 ** 8 %

Tämä johtuu alkuperäisen ASCII-merkistön muuntamisesta Pohjoismaiden normien mukaiseksi. Kantaluku ja eksponentti saavat olla vakioita tai muuttujia. Muuttuja voi olla liukuluku- tai kokonaislukumuodossa. Mikäli eksponentti on kokonaisluku, laskutoimituksen tarkkuus paranee, jos se myös esitetään kokonaislukumuodossa, siis %-merkin avulla. Negatiiviset luvut voidaan korottaa yksinomaan kokonaislukupotensseihin. Tämä on tärkeä muistaa.

Seuraava ohjelma laskee lausekkeen

$$y = x + 2x^2 + 3x^3$$

arvon eri x:n arvoilla.

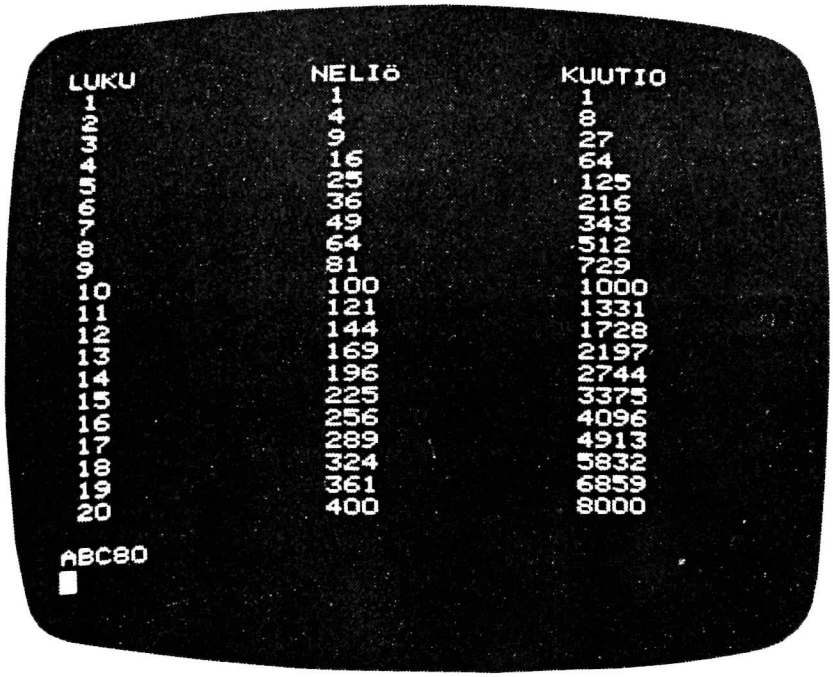
```
10 PRINT CHR$(12)
20 PRINT
30 PRINT
40 PRINT "X=";
50 INPUT X
60 LET Y=X+2*X**2%+3*X**3%
70 PRINT "Y=";Y
80 GOTO 20
```

Koska eksponentit ovat tässä esimerkissä kokonaislukuja, ne annetaan %-merkin avulla. Syötä muutamia eri arvoja x:lle ja tutki tulosta.

Voit myös jo aiemmin opitulla tavalla poistaa kokonaislukujen symbolit ja kokeilla, mitä tapahtuu.

Seuraava ohjelma tulostaa toisiinsa liittyvät arvot x, x^2 ja x^3 taulukkomuodossa.

```
10 PRINT CHR$(12)
20 PRINT "LUKU","NELIÖ","KUUTIO"
30 FOR X=1 TO 20
40 PRINT X,X**2%,X**3%
50 NEXT X
```



Taulukko, jossa on x:n potensseja

Funktiot

Kuten tämän luvun johdannossa mainittiin, ABC 80 -koneen BASIC-kielessä on koko joukko matemaattisia perusfunktioita, jotka helpottavat matemaattisten tehtävien ohjelmointia.

Matemaattiset funktiot:

Funktio	Merkitys	Huomautus
SIN(X)	sin x	x radiaaneina
COS(X)	cos x	
TAN(X)	tan x	
ATN(X)	arctan x	
LOG(X)	ln x	x:n luonnollinen logaritmi
LOG10(X)	lg x	x:n 10-kantainen logaritmi
EXP(X)	e^x	
SQR(X)	$\sqrt{x}$	x:n neliöjuuri

Matemaattiset erikoisfunktiot:

Funktio	Selitys
INT(X)	Suurin kokonaisluku, joka on pienempi tai yhtä suuri kuin x.
ABS(X)	x:n itseisarvo $\|x\|$.
SGN(N)	Antaa arvon —1 jos $x < 0$ 0 jos $x = 0$ 1 jos $x > 0$
FIX(X)	x:n kokonaisosa, [x].
PI	π eli luku 3,14159.
RND	Satunnaisluku väliltä 0—0,999999 reunat mukaanluettuina.

Osaan näistä olet jo tutustunutkin. Funktioita INT(X) ja RND käytettiin nimittäin jo 5. luvussa.

Useimpien funktioiden perässä on sulkumerkintä. Sulkeisiin merkitään ns. argumentti. Kaikki funktiot voivat sisältyä aritmeettisiin lausekkeisiin. Myös funktion argumenttina voi olla aritmeettinen lauseke.

Aritmeettinen lauseke

Aritmeettinen lauseke muodostetaan käyttämällä vakioita, muuttujia, funktioita, sulkeita ja aritmeettisia operaatioita. Lausekkeen muodostaminen noudattaa tavallisia matemaattisia sääntöjä.

Joitain esimerkkejä aritmeettisista lausekkeista.

Normaali kirjoitustapa	**BASIC-kielen kirjoitustapa**
$18 \cos x + x^2$	18*COS(X) + X**2%
$\ln (\arctan (\frac{x}{20})) + y (x + y)$	LOG((ATN(X/20))) + Y*(X + Y)
$\pi(A + \boxed{B}\ )$	PI*(A + INT(B))

Omien funktioiden määritteleminen

Lauseen DEF FN ... avulla voit määritellä omia funktioita. Esimerkiksi

```
10 DEF FNY(X)=SIN(X)*EXP(X+1)
20 DEF FNA(X)=TAN(X*SQR(X**3%))
```

määrittelee funktiot SIN (x) · e^{x+1} ja TAN (x · $\sqrt{x^3}$).

Sijoittaminen tapahtuu siis periaatteella DEF FNX (muuttuja) = aritmeettinen lauseke, jossa X saa olla mikä tahansa kirjain eli funktion nimisymboli.

Omien funktioiden määritteleminen on suuri etu, jos kyseessä olevaa funktiota on käytettävä useita kertoja ohjelmassa, esimerkiksi muuttujan eri arvoja vastaavien funktion arvojen laskemiseksi. Tällöin riittää funktiota vastaavan lausekkeen kirjoittaminen vain kerran. Ohjelma tulee näin lyhyemmäksi.

Esimerkki

Laske f(1) — f(0,5) + f(2)
kun f(x) = e^{-x2} + 1

Annetun laskutehtävän suorittava BASIC-ohjelma on seuraava:

```
10 DEF FNA(X)=EXP(-X**2%)+1
20 PRINT FNA(1)-FNA(.5)+FNA(2)
```

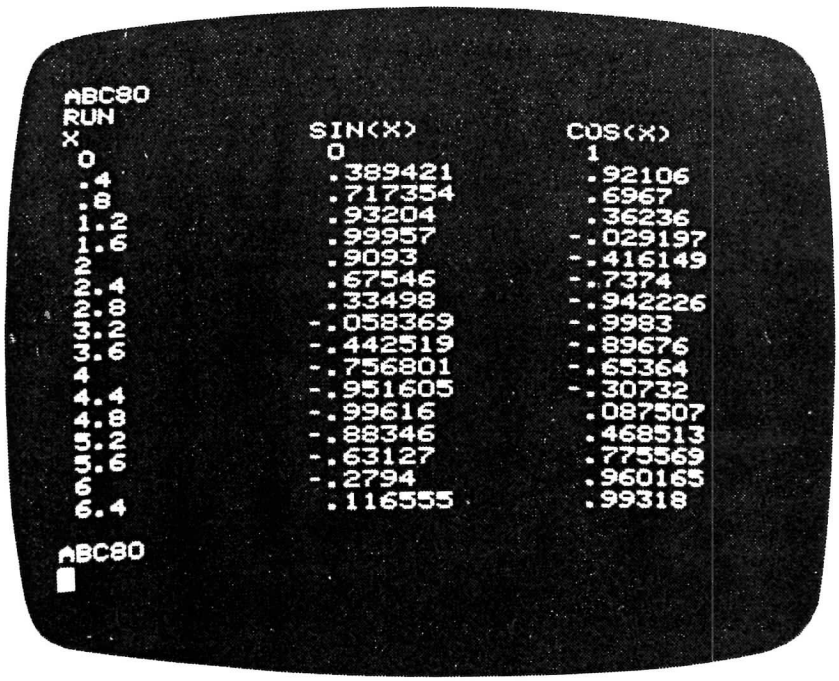
Funktion arvojen taulukointi

Potenssimerkintää koskevassa kirjan kohdassa olet jo kohdannut esimerkin, jossa funktion arvot kirjoitettiin taulukkomuodossa.

Seuraava ohjelma tuottaa samalla tavalla taulukoituina toisiinsa liittyvät arvot luvuille x, sin x ja cos x.

```
10 PRINT "X","SIN(X)","COS(X)"
20 FOR X=0 TO 6.4 STEP .4
30 PRINT X,SIN(X),COS(X)
40 NEXT X
```

Ohjelman rivillä 20 esiintyy FOR . . . TO . . . -käsky. Kun olet käyttänyt tätä käskyä aikaisemmin, on silmukkaan liittyneen askeleen pituus ollut aina 1, mikä merkitsee sitä, että silmukkamuuttujan arvoa on lisätty 1:llä jokaisella silmukan kierroksella. Käskyn STEP lisäämisellä voidaan askeleen pituudeksi valita mikä tahansa arvo. Tässä ohjelmassa suurennetaan silmukkamuuttujan X arvoa luvulla 0,4 joka kerta, kun rivien 20, 30 ja 40 määrittelemä silmukka suoritetaan. Ohjelman ajo antaa seuraavan tulostuksen.



Funktioiden arvot taulukoituina

Funktioiden sin x ja cos x arvot annetaan aina kuuden numeron tarkkuudella. Merkityksettömiä nollia ei tulosteta näyttöön. Monissa tapauksissa on kuitenkin tarpeetonta esittää tulosta kuusinumeroisena. Funktion INT avulla voidaan laskutoimituksen tai funktiotoiminnon tulos aina pyöristää haluttuun numeromäärään.

Pyöristäminen

INT(X) antaa suurimman kokonaisluvun, joka on pienempi tai yhtä suuri kuin X.

Olet jo 4. luvussa käyttänyt INT-funktiota pyöristäessäsi laskutoimituksen tuloksen täysiksi markoiksi.

Esimerkki

```
INT(610.991) = 610
INT(18) = 18
INT(-7.8) = -8          (-8 on pienempi kuin -7,8)
```

Heikkoutena on, että pyöristäminen tapahtuu lähinnä pienimpään kokonaislukuun. Normaalisti pitäisi pyöristämisen tapahtua lähimpään kokonaislukuun.

Esimerkiksi

```
18,588 pyöristetään luvuksi 19
18,493 pyöristetään luvuksi 18
```

Tämä oikea pyöristäminen tapahtuu siten, että lukuun lisätään ensin 0,5, jonka jälkeen summasta otetaan kokonaisosa.

Esimerkiksi

```
INT(18.588 + .5) = 19
INT(18.493 + .5) = 18
```

Jos sen sijaan haluat pyöristää yhden desimaalin tarkkuuteen, kerro luku ensin 10:llä, lisää 0,5 ja jaa INT-funktion arvo lopuksi 10:llä.

Esimerkiksi

```
INT(10 * 18.588 + .5)/10 = 18.6
INT(10 * 18.493 +.5)/10 = 18.5
```

Kahden desimaalin tarkkuuteen pyöristettäessä kerrotaaan luku ensin 100:lla ja INT-funktion arvo jaetaan lopuksi 100:lla.

Esimerkiksi

```
INT(100 * 18.588 + .5)/100 = 18.59
INT(100 * 18.493 + .5)/100 = 18.49
```

Saamme seuraavan yleisen kaavan pyöristystehtäviä varten:

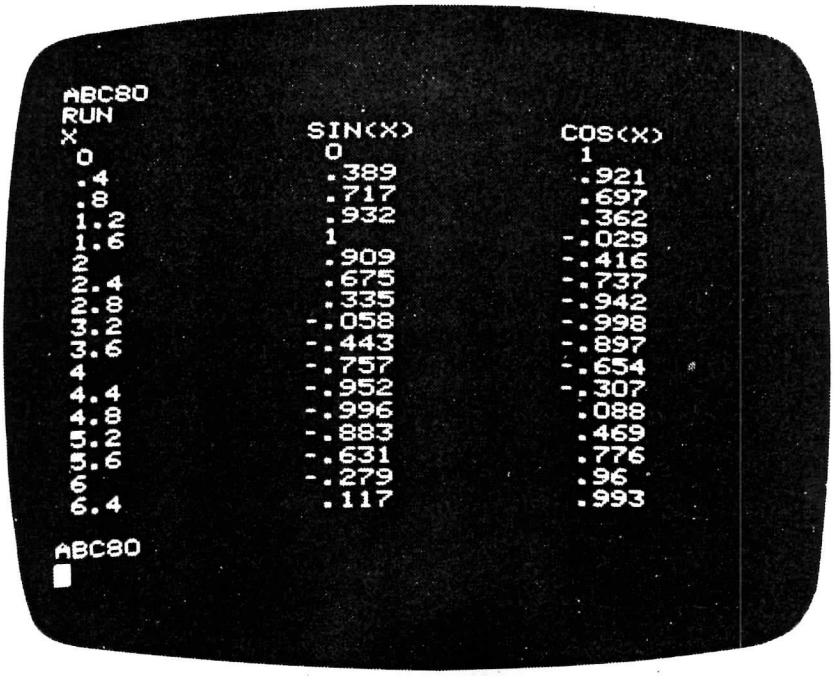
INT(10 * * D % * X + .5)/10 * * D %

Se pyöristää luvun X muotoon, jossa on D desimaalia.

Muunnamme nyt aikaisempaa ohjelmaamme siten, että toisiinsa liittyvät arvot x, sin x ja cos x tulostetaan edelleen taulukkomuodossa mutta nyt kolmen desimaalin tarkkuuteen pyöristettyinä.

```
10 PRINT "X","SIN(X)","COS(X)"
20 FOR X=0 TO 6.4 STEP .4
30 PRINT X,INT(1000*SIN(X)+.5)/1000,
INT(1000*COS(X)+.5)/1000
40 NEXT X
```

Ohjelman ajo tuottaa seuraavan näytön



Funktioiden arvot halutulla tarkkuudella

Funktion kuvaajan piirtäminen käskyllä PRINT TAB

Kun liitämme tavallisen tulostuskäskyn PRINT perään käskyn TAB(K), saamme määrätyksi, mihin kohtaan riviä tulostus muodostuu. Muuttujan K arvon kokonaislukuosa, missä K voi olla myös aritmeettinen lauseke, ilmaisee, mistä rivin kohdasta näyttöön tuleva merkkijono tai luku alkaa. K:n mahdollisten arvojen kokonaisosan tulee olla välillä 0—39.

Esimerkiksi

```
10 PRINT TAB(12);"MATEMATIIKKA"
```

antaa seuraavan näytön

```
           MATEMATIIKKA
           ↑
Rivin kohta 12
```

Käskyn TAB avulla voimme myös piirtää yksinkertaisia käyriä näyttöön.

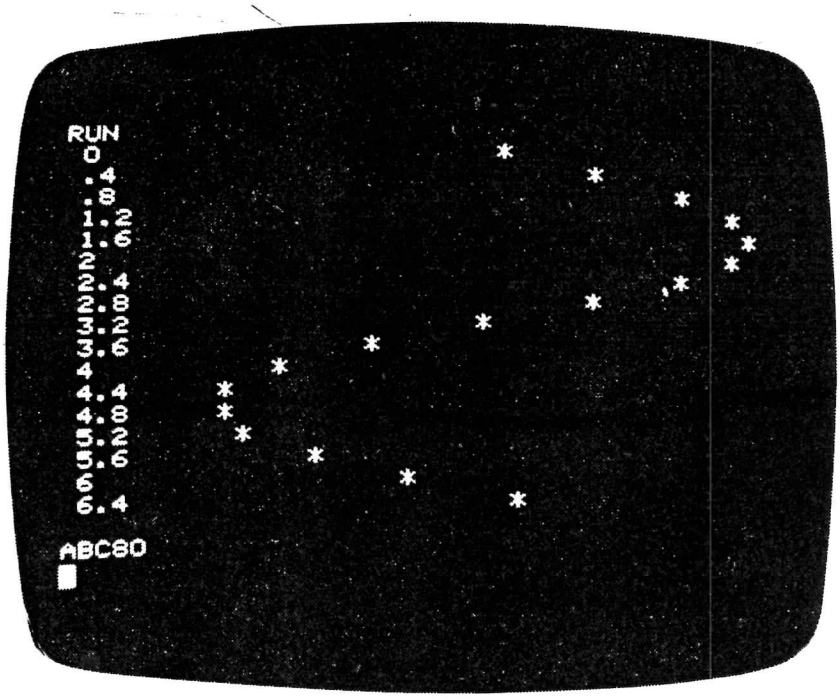
Valitsemme esimerkiksi funktion y = sin x kuvaajan piirtämisen. Ohjelmaan sisällytetään PRINT-lause, joka muotoillaan niin, että näytön vasempaan reunaan kirjoitetaan x-arvot, joita vastaavat kuvaajan pisteet piirretään oikealle puolelle. Käskyä TAB käytämme ilmoittamaan sen paikan, johon esimerkiksi merkki "*" kirjoitetaan vastaamaan funktion y-arvoa. Sen lisäksi on funktion arvot mitoitettava eli jaoteltava sopivasti. Se tapahtuu siten, että arvot kerrotaan tietyllä vakiolla ja tuloon lisätään luku siten, että käytettävissä oleva näyttöruudun osa tulee peitetyksi mahdollisimman hyvin. Valitsemalla K = 24 + 15 Y käskyssä TAB(K), saadaan näyttö vaakasuunnassa asemien 9 ja 39 väliin, koska sin x:n arvot vaihtelevat välillä —1 ja +1. Täten näytön vasempaan reunaan jää 9:n merkin levyinen tila käytettäväksi muuttujan x arvoille.

Funktion y = sin x kuvaajan piirtävä ohjelma on seuraavan näköinen:

```
10 FOR X=0 TO 6.4 STEP .4
20 LET Y=SIN(X)
30 PRINT X;TAB(24+15*Y);"*"
40 NEXT X
```

Rotkan
Kaupunginkirjasto

Ohjelman ajo antaa seuraavan tuloksen:



Sinikäyrä TAB-funktion avulla piirrettynä

Ota huomioon, että ajateltu y-akseli osoittaa näytössä oikealle, kun taas x-akseli osoittaa alaspäin.

Funktion kuvaajan piirtäminen käskyllä PRINT CUR

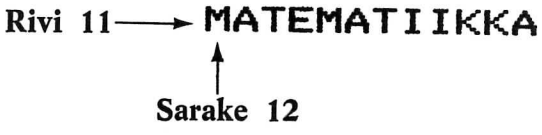
Valitettavasti eivät koordinaattiakselit suuntautuneet meille tutulla tavalla, kun piirsimme käyrän käskyn PRINT TAB avulla.

Käskyn CUR(R, S) saadaan tulostus alkamaan halutulta riviltä R ja sarakkeesta S. R ja S voivat olla myös aritmeettisia lausekkeita. Luvun R kokonaisosan tulee olla lukujen 0 ja 23 välillä ja vastaavasti luvun S kokonaisosan lukujen 0 ja 39 välillä.

Esimerkiksi

```
10 PRINT CUR(11,12);"MATEMATIIKKA"
```

antaa seuraavan tulostuksen:



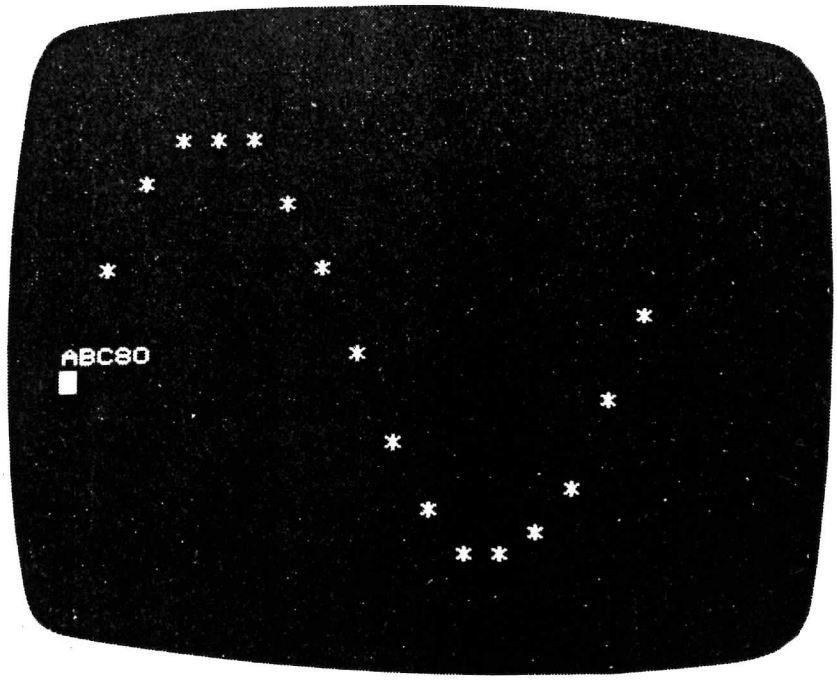
Käskyä CUR voidaan käyttää piirtämiseen esimerkiksi merkin "*" avulla siten, että sen paikka määrätään luvuilla R ja S, jotka vastaavat koordinaatteja x ja y. Näyttöä käytetään jälleen parhaiten hyväksi jaotuksen eli skaalauksen avulla.

Funktion y = sin x kuvaaja voidaan esittää tutulla tavalla käyttämällä seuraavaa ohjelmaa:

```
10 PRINT CHR$(12)
20 FOR X=0 TO 6.4 STEP .4
30 LET Y=SIN(X)
40 PRINT CUR(12-10*Y,5*X);"*"
50 NEXT X
```

Mieti, miksi skaalaus rivillä 40 on valittu juuri näin. Miksi rivin 40 muuttaminen vaikuttaa tulostukseen. Kokeile eri skaalauksia. Palauta kuitenkin alkuperäinen jaotus ennen kuin jatkat seuraavia muutoksia.

Ohjelman ajo antaa seuraavan tuloksen:

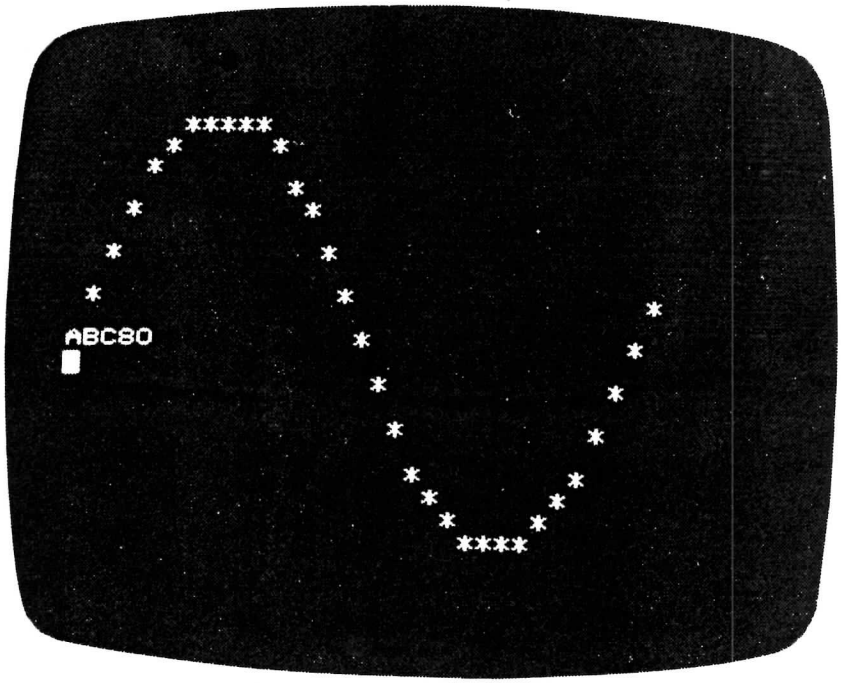


Sinikäyrä funktion CUR avulla piirrettynä

Muuta nyt ohjelmassa oleva askeleen pituus arvosta 0,4 arvoksi 0,2 eli muuta rivi 20 muotoon

```
20 FOR X=0 TO 6.4 STEP .2
```

Tällöin antaa ohjelman ajo tuloksen:



Askeleen pituus vaikuttaa käyrän muotoon

Kuten yllä olevasta tuloksesta näkyy, tulevat sinikäyrän huiput latteiksi. Tämä johtuu tässä tulostustarkkuuden riittämättömyydestä. Tutki tässä mielessä myös harjoitustehtävää 9.7, joka osoittaa, että tarkkuus tällä menetelmällä ei riitä täysin oikeasti muotoutuvan käyrän saamiseksi.

Tarkemmin kaartuvien käyrien ja kuvioiden aikaansaamiseksi voidaan ABC 80 -koneessa käyttää ns. graafista käyttömuotoa. Tällöin jaetaan jokainen merkkipaikka kuudeksi graafiseksi pisteeksi. Seuraavassa eli 10. luvussa tutustumme graafiseen tulostukseen ja toteamme samalla, kuinka mm. edellä esitetyn käyrän näyttöä voidaan huomattavasti parantaa.

Yhteenveto

Käsky	Selitys	Esimerkki
`FOR...TO... STEP...`	Tekee mahdolliseksi määritellä askeleen pituus FOR...TO... -lauseessa	`10 FOR X=1 TO 10 STEP 2`
`DEF FN... (...)`	Omien funktioiden määritteleminen.	`10 DEF FNA(X,Y) =X*X+Y`

Funktio	Selitys	Esimerkki
SIN(X)	sin x, x radiaaneina	10 Y=SIN(X)
COS(X)	cos x, x radiaaneina	20 Y=COS(5*X)
TAN(X)	tan x, x radiaaneina	30 Y=TAN(X+2)
ATN(X)	arctan x, x radiaaneina	40 Y=ATN(.5)
LOG(X)	x:n luonnollinen logaritmi	50 Y=LOG(756)
LOG10(X)	x:n 10-kantainen logaritmi	60 Y=LOG10(1000%)
EXP(X)	e^x	70 Y=EXP(2)
SQR(X)	x:n neliöjuuri, $\sqrt{x}$	80 Y=SQR(A**2+ B**2)
INT(X)	Suurin kokonaisluku, joka on pienempi tai yhtä suuri kuin x	90 Y=INT(Z)
FIX(X)	x:n kokonaisosa, [x]	100 Y=FIX(Z)
ABS(X)	x:n itseisarvo, $\|x\|$	110 Y=ABS(-6)
SGN(X)	-1, jos $x < 0$ 0, jos $x = 0$ 1, jos $x > 0$	120 Y=SGN(X)
PI	π, eli 3,14159	130 X=PI*R**2
TAB(S)	Siirtää rivillä osoittimen asemaan S. (Esiintyy vain PRINT-lauseissa.)	140 PRINT TAB(10); "SARAKE 10"
CUR(R,S)	Siirtää osoittimen rivin R sarakkeeseen eli asemaan S näytössä. Esiintyy vain PRINT-lauseissa.)	150 PRINT CUR(11, 12); "KESKELLE KUVARUUTUA"

Harjoituksia

9.1 Tee ohjelma, joka laskee lausekkeen $x^5 + x^2 + 2$ arvon. Käytä INPUT-lausetta eri x-arvojen syöttöön.

9.2 Kirjoita ohjelma, joka tulostaa taulukkomuodossa arvot x, ln x ja e^x, kun x saa kokonaislukuarvot välillä 10 ja 19.

9.3 Muuta edellisen tehtävän muuttujan x askelpituus siten, että se on 0,5.

9.4 Kirjoita ohjelma, jossa INPUT-lauseessa määritellään muuttujat A ja B sekä lasketaan lausekkeen $C = \sqrt{A^2 + B^2}$ arvo.

9.5 Tee ohjelma, jossa DEF FN-käskyn avulla määritellään funktio $f(x) = \sqrt{x} \cdot e^{-x2}$ sekä lasketaan sen jälkeen f(x) kun $x = 1.7$, 2.25 ja 2.5.

9.6 Laske tekemäsi ohjelman avulla neliöjuuri syötetystä luvusta kahden desimaalin tarkkuuteen pyöristettynä.

9.7 Tee käyttäen mm. PRINT CUR -käsky-yhdistelmää hyväksesi ohjelma, joka tulostaa graafisesti funktion y = arc tan x, kun x vaihtelee välillä −10 ja +10. Yritä käyttää kuvaruutua niin tehokkaasti kuin mahdollista.

9.8 Funktio ABS(X) antaa arvon x itseisarvon. Muunna sivun 111 ohjelmaa siten, että rivillä 40 vaihdat muuttuja-arvon y arvoksi ABS(Y). Millainen on tulos ajaessasi ohjelman. Selvitä itsellesi, mitä tapahtuu.

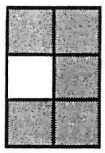
KAPPALE 10

Tietokone esittää kuvioita

ABC80

Graafinen tulostusmuoto

ABC 80 -tietokoneen avulla voit piirtää monitoriin kuvaajia sekä graafisia kuvioita ja diagrammeja käyttämällä hyväksi sitä ominaisuutta, että useimmilla ASCII-merkeillä on vastaava graafinen merkki. Kutakin niistä vastaa tietty valkeiden ruutujen muodostama kuvio seuraavan kaavion avulla.



Yllä on esitetty merkkiä "ä" vastaava graafinen merkki.

ASCII-taulukko vastaavin graafisin merkein

Seuraavalla sivulla oleva taulukko ilmoittaa eri ASCII-merkkien graafisen vastineen. Kuten huomaat, ei isoilla kirjaimilla ole vastaavaa graafista merkkiä. Tämä merkitsee sitä, että graafinen käyttömuoto ei vaikuta isoihin kirjaimiin, vaan ne tulostuvat sellaisinaan tässäkin tulostusmuodossa. Graafiset merkit vastaavat ns. viewdata-normeja.

Graafisen tulostusmuodon erikoismerkit

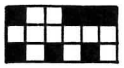
Se, käytetäänkö näytössä normaalia tapaa vai graafisia merkkejä, valitaan kahdella erikoismerkillä, joilla ohjataan graafista tulostusmuotoa.

CHR¤(151) antaa merkin "ALOITA GRAFIIKKA"
CHR¤(135) antaa merkin "LOPETA GRAFIIKKA"

Tavallisesti merkit kirjoitetaan tietenkin normaalilla tavalla, mutta kun jollekin riville kirjoitetaan "näkymätön" merkki CHR¤(151), muuttuu kaikki sen jälkeinen näyttö merkkien graafista muotoa vastaavaksi. Seuraava esimerkki selventää kyseistä periaatetta.

```
10 PRINT "123ABC";CHR¤(151);"123ABC"
```

Ajon jälkeinen kuvaruutu on

123ABC  ABC

Kun haluamme palauttaa normaalin tulostusmuodon, käytämme edellä esitettyä käskyä "LOPETA GRAFIIKKA" eli CHR¤(135).

Koodi	M	G	Koodi	M	G	Koodi	M	G	Koodi	M	G
32	Space		56	8		80	P	P	104	h	
33	!		57	9		81	Q	Q	105	i	
34	"		58	:		82	R	R	106	j	
35	#		59	;		83	S	S	107	k	
36	¤		60	<		84	T	T	108	l	
37	%		61	=		85	U	U	109	m	
38	&		62	>		86	V	V	110	n	
39	'		63	?		87	W	W	111	o	
40	(		64	É	É	88	X	X	112	p	
41	)		65	A	A	89	Y	Y	113	q	
42	*		66	B	B	90	Z	Z	114	r	
43	+		67	C	C	91	Ä	Ä	115	s	
44	,		68	D	D	92	Ö	Ö	116	t	
45	-		69	E	E	93	Å	Å	117	u	
46	.		70	F	F	94	Ü	Ü	118	v	
47	/		71	G	G	95	–	–	119	w	
48	0		72	H	H	96	é		120	x	
49	1		73	I	I	97	a		121	y	
50	2		74	J	J	98	b		122	z	
51	3		75	K	K	99	c		123	ä	
52	4		76	L	L	100	d		124	ö	
53	5		77	M	M	101	e		125	å	
54	6		78	N	N	102	f		126	ü	
55	7		79	O	O	103	g		127	■	

ASCII-merkit merkkimuodossa (M) ja grafiikkamuodossa (G) tulkittuina desimaalikoodin mukaan.

Grafiikkamerkit vastaavat mm. englantilaisen VIEWDATA- ja suomalaisen TELSET -järjestelmien merkkejä.

```
10 PRINT CHR¤(151);"123";CHR¤(135);"567"
```

Yllä oleva ohjelmarivi alkaa graafiseen tulostusmuotoon siirtymisellä. Se koskee riviä 10 niin kauan, kunnes käsky "LOPETA GRAFIIKKA" muuttaa tulostusmuodon takaisin normaaliksi. Niinpä jono "123" tulee kirjoitetuksi graafisessa muodossa. Sen sijaan rivin lopussa oleva 567 kirjoitetaan normaalisti lukumerkintänä.

Ajon jälkeinen monitorissa on tulostus

Käskyn CHR¤ (135) sijainti

Käskyn CHR¤(151) sijainti

Koko käyttö graafisessa tulostusmuodossa

Koko näyttö voidaan saattaa graafiseen muotoon seuraavalla ohjelmalla:

```
10 PRINT CHR¤(12)
20 FOR I=0 TO 23
30 PRINT CUR(I,0);CHR¤(151);
40 NEXT I
```

Tämä ohjelma aiheuttaa sen, että käsky "ALOITA GRAFIIKKA" tulee kirjoitetuksi ensimmäisenä merkkinä jokaisella kuvaruudun 24 rivillä.

Vain mielikuvitus asettaa rajoja

Kun olet edellä esitetyn ohjelman avulla saattanut koko näytön graafiseen muotoon, voit sijoittaa osoittimen sopivaan paikkaan käskyn CUR(R, K) avulla ja kirjoittaa siten haluamiasi graafisia kuvioita. Lisää seuraavat rivit, niin saat ohjelman, joka piirtää näyttöön kuusi ruutua.

```
50 FOR A=5 TO 30 STEP 5
60 PRINT CUR(10,A);"wä"
70 NEXT A
```

ABC 80 -tietokoneen mukana seuraavassa esittelykasetissa on useita ohjelmaesimerkkejä, jotka kuvaavat graafisten merkkien erilaisia käyttömahdollisuuksia.

Jos koko näyttöruutu saatetaan graafiseen tulostusmuotoon, saadaan 72 graafista riviä (R) numeroituina 0—71 ja 78 graafista paikkaa eli saraketta (K) numeroituina 2—79 jokaiselle riville. Käskyllä SETDOT R, K saadaan sytytetyksi mikä tahansa graafinen piste. Tarkista, että seuraava ohjelma sytyttää näyttöruutuun neljä pistettä.

```
10 PRINT CHR¤(12)
20 FOR I=0 TO 23
30 PRINT CUR(I,0);CHR¤(151);
40 NEXT I
50 SETDOT 30,30
60 SETDOT 30,40
70 SETDOT 40,30
80 SETDOT 40,40
```

Merkki CHR¤(151) eli "ALOITA GRAFIIKKA" on jatkuvasti jokaisen rivin ensimmäisenä merkkinä. Kuten olemme aikaisemmin osoittaneet, tämä merkki on näytössä näkymätön, mutta sen kohdalle ei voi kirjoittaa mitään muuta. Sen vuoksi voit käyttää kullakin rivillä vain paikkoja 2—79.

Lisäksi on olemassa vielä yksi käsky, jota voidaan käyttää vain graafisessa muodossa, nimittäin CLRDOT R, K, joka sammuttaa graafisen pisteen. Sen lisäksi on olemassa päättely- eli ehtofunktio DOT(R, K), joka testaa, onko tietty piste sytytetty vai ei. Seuraava ohjelmarivi sammuttaa pisteen R, K, jos se on sytytetty, mutta sytyttää sen, jos se on sammutettu.

```
110 IF DOT(R,K) THEN CLRDOT R,K ELSE
SETDOT R,K
```

Lause IF ... THEN ... ELSE ... merkitsee sitä, että jos sanan IF jäljessä oleva ehto on täytetty, suoritetaan sanan THEN jälkeinen käsky. Jos ehto sitävastoin ei ole täytetty, suoritetaan sanan ELSE jäljessä oleva käsky.

Lisää nyt rivit

```
90 FOR R=20 TO 50
100 FOR K=20 TO 50
120 NEXT K
130 NEXT R
```

Koko ohjelma näyttää nyt seuraavalta:

```
10 PRINT CHR$(12)
20 FOR I=0 TO 23
30 PRINT CUR(I,0);CHR$(151);
40 NEXT I
50 SETDOT 30,30
60 SETDOT 30,40
70 SETDOT 40,30
80 SETDOT 40,40
90 FOR R=20 TO 50
100 FOR K=20 TO 50
110 IF DOT(R,K) THEN CLRDOT R,K ELSE
SETDOT R,K
120 NEXT K
130 NEXT R
```

Aja ohjelma ja selvitä itsellesi sen toiminta!

Käyrän piirtäminen paremmalla tarkkuudella

Edellisessä luvussa "piirsimme" käyriä funktion CUR(R, S) avulla. Maksimaalinen tulostustarkkuus oli 24 paikkaa y-suunnassa ja 40 paikkaa x-suunnassa. Jos sen sijaan käytämme funktiota SETDOT R, K, saamme merkittävästi paremman tarkkuuden, nimittäin 72 paikkaa y-suunnassa ja 78 x-suunnassa.

Kun käytetään käskyä SETDOT R, K, täytyy näytön jaotuksen luonnollisesti tapahtua vastaavalla tavalla kuin käytettäessä funktiota CUR(R, S) käyrien piirtämiseksi.

Sinikäyrä graafisen tulostuksen avulla

Alla olevan ohjelman avulla voidaan piirtää sinikäyrä.

```
10 PRINT CHR$(12)
20 FOR I=0 TO 23
30 PRINT CUR(I,0);CHR$(151);
40 NEXT I
50 FOR X=0 TO 6.4 STEP .1
60 R=35-30*SIN(X)
70 S=2+10*X
80 SETDOT R,S
90 NEXT X
100 GOTO 100
```

Aja ohjelma. Ohjelma sisältää päättymättömän silmukan, ja niinpä ajo on katkaistava näppäimillä CTRL ja C .

Rivinumero	Rivin toiminto
10	Tyhjentää näytön.
20—40	Koko näyttö saatetaan graafiseen tulostusmuotoon.
50, 90	Määrittelee silmukan, jossa silmukkamuuttuja x saa arvot välillä 0—6,4 askeleen pituuden ollessa 0,1.
60	Jaottaa sin x:n arvot sopivasti y-suunnassa.
70	Jaottaa x:n arvot sopivasti x-suunnassa.
80	Tuottaa graafisen pisteen.
100	Aiheuttaa hypyn takaisin riville 100, eli antaa päättymättömän silmukan, joka eliminoi tekstin ABC80 näytöstä.

Muista edellisen jaotuksesta, että käytettävissäsi on 72 × 80 ruudukko ja että funktio sin (x) saa arvot −1—+1!

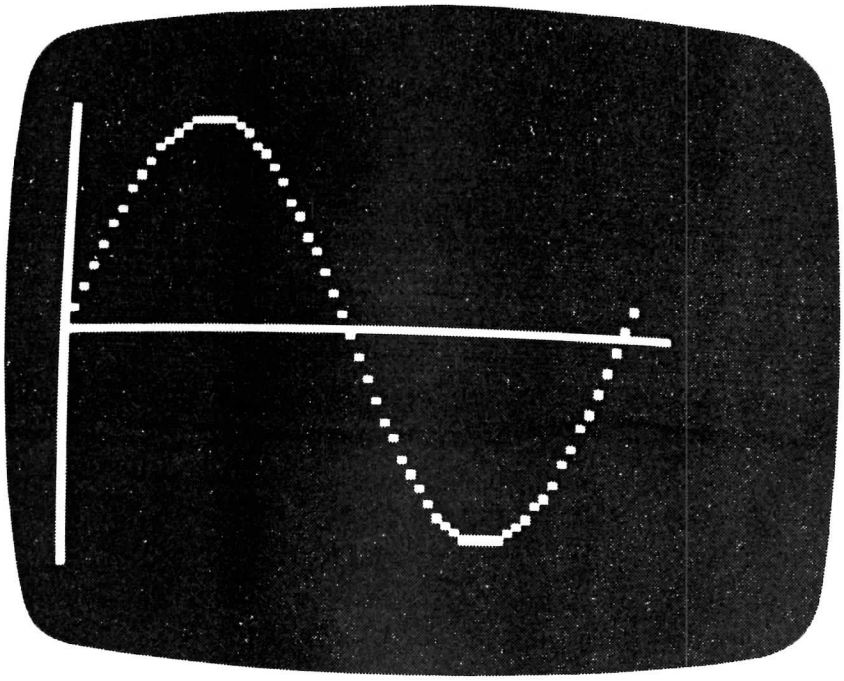
Sinikäyrä ja koordinaattiakselit

Jos käyrän tulostusmuoto halutaan varustaa akseleilla, on helppoa lisätä ohjelmaan kaksi silmukkaa, jotka piirtävät x- ja y-akselit.

```
41 FOR R=3 TO 67
42 SETDOT R,2
43 NEXT R
44 FOR S=2 TO 76
45 SETDOT 35,S
46 NEXT S
```

Muista, että millään rivillä ei saa nyt kirjoittaa kahteen ensimmäiseen graafiseen paikkaan. Niillähän sijaitsee näkymätön merkki "ALOITA GRAFIIKKA".

Ohjelman ajon jälkeen saadaan ruutuun seuraava kuva

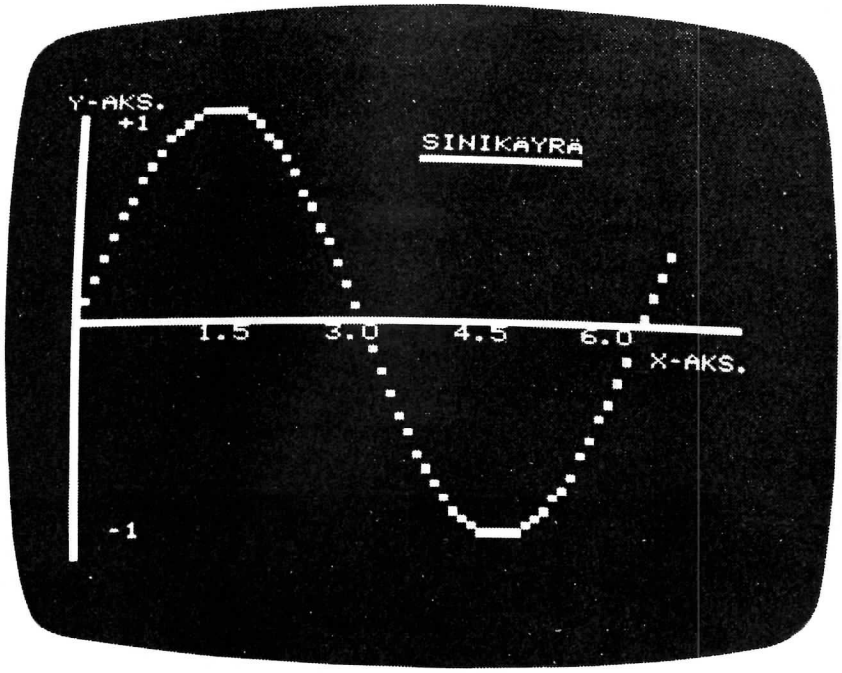


Sinikäyrä varustettuna akseleilla

Sinikäyrä ja asteikolla varustetut koordinaattiakselit

Jos niin halutaan, voidaan akselit varustaa sopivilla näyttömerkinnöillä asteikoille kuten myös kirjoittaa näyttöön selventävää tekstiä.

Alla olevassa kuvassa esimerkki



Sinikäyrä ja asteikolla varustetut akselit

Edellä olevan näytön aikaansaamiseksi on ohjelmaan lisätty seuraavat rivit:

```
100 REM **TEKSTI GRAFIIKASSA**
110 PRINT CUR(1,0);"Y-AKS.";CHR$(151)
120 PRINT CUR(13,32);CHR$(135);"X-AKS."
130 PRINT CUR(3,20);"SINIKÄYRÄ"
140 PRINT CUR(4,20);"##########"
150 PRINT CUR(12,7);CHR$(135);"1.5
3.0     4.5     6.0";CHR$(151)
160 PRINT CUR(2,2);CHR$(135);"+1";CHR$
(151)
170 PRINT CUR(21,2);CHR$(135);"-1";CHR$
(151)
180 GOTO 180
```

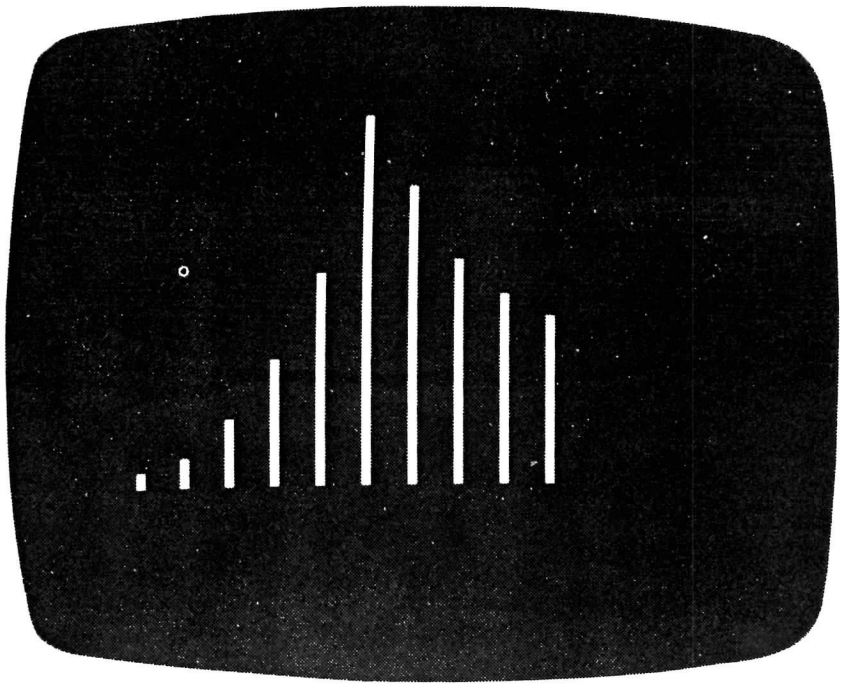
Mieti tarkoin näiden ohjelmarivien sisältöä. Kun joudumme sekoittamaan keskenään normaalia ja graafista näyttöä, meille on paljon hyötyä ns. "grafiikkakartasta", joka on tämän luvun lopussa.

Pylväsdiagrammi

Alla oleva ohjelma esittää, kuinka käskyn SETDOT R, S avulla voidaan piirtää pylväsdiagrammi. Tehtävänä on, että diagrammi esittää kymmenen arvoa lukujen 0 ja 50 välillä.

```
10 PRINT CHR$(12);"SYÖTÄ SISÄÄN 10
LUKUA VÄLILTÄ 0 - 50 "
20 PRINT " (EROTA NE 'RETURN'
-NÄPPÄIMELLÄ)"
25 PRINT : PRINT
30 FOR N=1 TO 10
40 PRINT N"=";
45 INPUT A(N)
50 NEXT N
60 PRINT CHR$(12)
70 FOR I=0 TO 17
80 PRINT CUR(I,0);CHR$(151);
90 NEXT I
100 FOR S=10 TO 55 STEP 5
110 X=X+1
120 FOR R=50 TO 50-A(X) STEP -1
130 SETDOT R,S
140 NEXT R
150 NEXT S
160 GOTO 160
```

Ajon tulos voi näyttää seuraavalta:



Pylväsdiagrammi

Varusta ohjelma kommentoivilla teksteillä lisäämällä sopivia PRINT-lauseita.

Yhteenveto

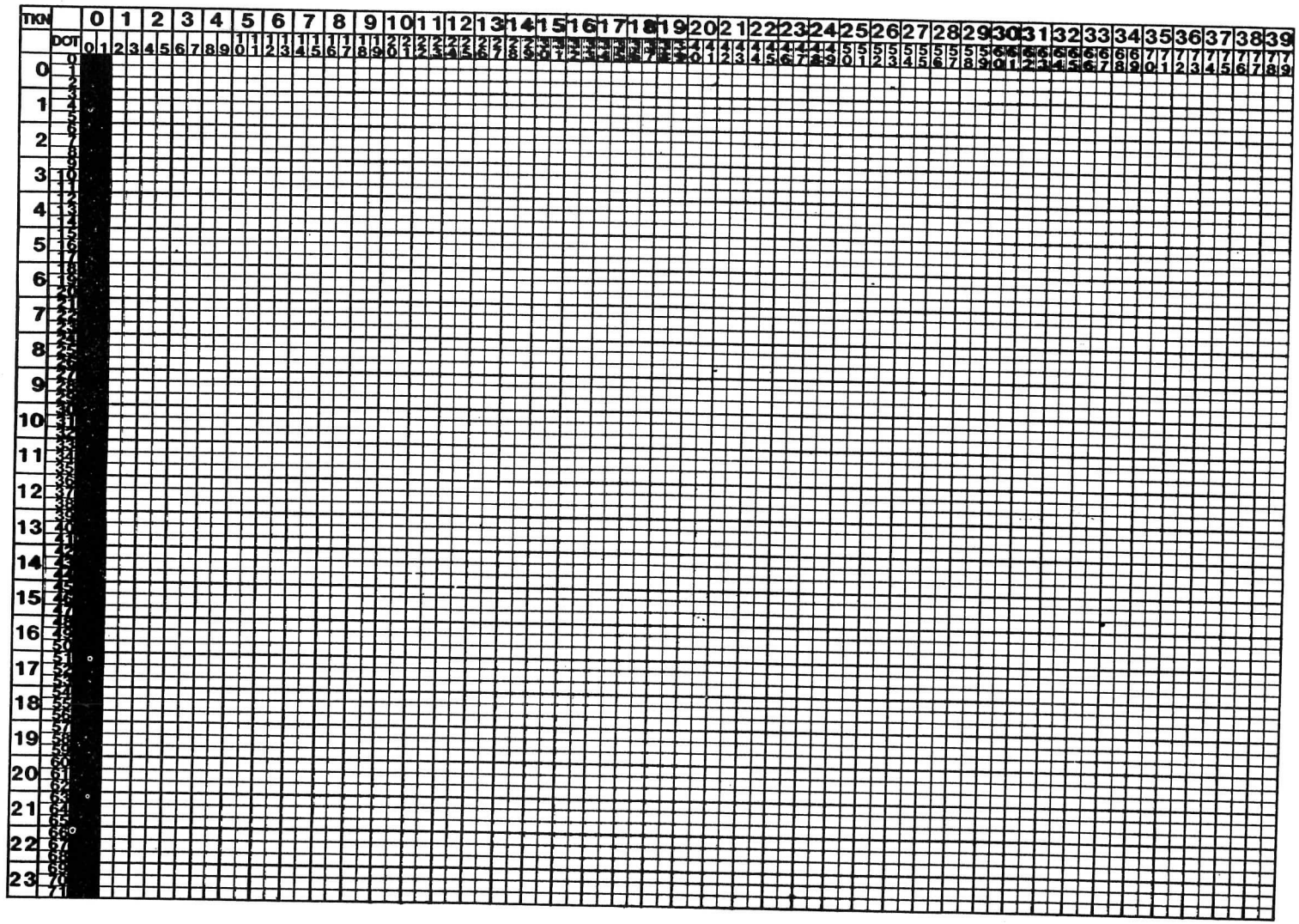
Käsky	Selitys	Esimerkki
IF...THEN... ELSE...	Jos sanan IF jäljessä oleva ehto on tosi, suoritetaan sanan THEN jäljessä oleva käsky. Muussa tapauksessa suoritetaan sanan ELSE jäljessä oleva käsky.	`10 IF A=B THEN PRINT A ELSE GOTO 300`
	Jos sanan THEN tai ELSE jäljessä on vain rivinumero, hypätään tälle riville.	`20 IF A=B THEN 90 ELSE 200`
	ELSE voidaan myös jättää pois.	`30 IF A=B THEN 50`

SETDOT R,S	Sytyttää näyttöruutuun graafisen pisteen. (Rivin täytyy olla asetettuna graafiseen näyttömuotoon.) $0 \leq R \leq 72$ $2 \leq S \leq 79$	40 SETDOT 30, 40
CLRDOT R,S	Sammuttaa näyttöruudun graafisen pisteen.	50 CLRDOT 30, 40

Ehtoilmaus	Selitys	Esimerkki
DOT(R,S)	Antaa arvon TOSI, jos graafinen piste R, S on sytytettynä. Muussa tapauksessa arvo on EPÄTOSI.	30 IF DOT(30, 40) THEN CLRDOT 30,40 SETDOT ELSE 30,40

Erikoismerkki	Selitys	Esimerkki
CHR$(151)	"ALOITA GRAFIIKKA" aloittaa rivillä graafisen näyttömuodon.	10 PRINT CHR$ (151);"1234"
CHR$(135)	"LOPETA GRAFIIKKA" palauttaa rivillä normaalin näyttömuodon.	20 PRINT CHR$ (151);"1234"; CHR$(135);"1"

ABC80 VIDEOKARTTA OHJELMA..... .....



= paikat merkille CHR ¤ (151) = "ALOITA GRAFIIKKA"

Harjoituksia

10.1 Tee ohjelma, jossa ensin asetat koko kuvaruudun graafiseen tilaan ja tämän jälkeen piirrät n. 6 cm pitkän miinusmerkin keskelle kuvaruutua. Kokeiltuasi sitä ja todettuasi sen toimivan muunna ohjelmaa edelleen siten, että ruutuun tulostuu vastaavankokoinen plusmerkki.

10.2 Alla oleva ohjelma tulostaa ASCII-merkit numerosta 32 numeroon 127. Täydennä ohjelmaa siten, että samalle riville tulostuu myös kunkin ASCII-merkin graafinen vastinesymboli.

```
10 FOR A=32 TO 127
20 PRINT
30 PRINT CHR$(A)
40 FOR X=1 TO 1000 : NEXT X
50 NEXT A
```

10.3 Muunna sivun 123 histogrammiohjelmaa siten, että pylväät tulostetaan mustina valkoisella pohjalla.

10.4 Kirjoita ohjelma, joka piirtää ison ikkunan kehykset kuvaruutuun SET DOT -käskyä käyttäen. Yritä saada aikaan niin suuret kehykset kuin mahdollista.

10.5 Harjoitustehtävän 10.4 ohjelmaratkaisun laajennettu versio on esitetty alla. Yritä analysoida ohjelman kulku, kun olet koeajanut sen.

```
130 X1=1 : Y1=1 : X=2 : Y=0
140 FOR A=1 TO 5500
150 X=X+X1 : Y=Y+Y1 : SETDOT Y,X
160 IF X=79 THEN X1=-1
170 IF X=2 THEN X1=1
180 IF Y=71 THEN Y1=-1
190 IF Y=0 THEN Y1=1
200 NEXT A
210 FOR A=1 TO 5500
220 X=X+X1 : Y=Y+Y1 : CLRDOT Y,X
230 IF X=79 THEN X1=-1
240 IF X=2 THEN X1=1
250 IF Y=71 THEN Y1=-1
260 IF Y=0 THEN Y1=1
270 NEXT A : GOTO 10
```

10.6 Alla oleva ohjelma esittää yhden mahdollisuuden saada aikaan yksinkertaista liikettä kuvaruudulla.

```
10 PRINT CHR$(12)
20 FOR I=0 TO 23
30 PRINT CUR(I,0);CHR$(151);
40 NEXT I
50 FOR X=20 TO 40
60 SETDOT X,29
70 SETDOT X,61
80 NEXT X
90 S=30 : A=1
100 R=30 : S=S+A
110 SETDOT R,S
120 FOR X=1 TO 20 : NEXT X
130 CLRDOT R,S
140 FOR X=1 TO 20 : NEXT X
150 IF S=60 THEN A=-1
160 IF S=30 THEN A=1
170 GOTO 100
```

Miten liikkuvan pisteen nopeutta voidaan muuttaa?

10.7 Yritä kehittää ohjelma, jota voidaan käyttää murtohälyttimenä kalakauppias Muikun liikkeessä. Ohjelman pitää tulostaa kuvaruutuun seuraava teksti:

Näppäile yhdistelmä XQW,
niin kassakaappi aukeaa.

Kun murtovaras sitten yöllä tulee ja lähestyy ABC-tietokonetta, ei hän tietenkään voi olla painamatta näppäintä X. Ohjelman on tällöin käynnistettävä äänigeneraattori ja kirjoitettava sopiva murtoja ehkäisevä hälytysteksti kehystettynä kuvaruutuun.

KAPPALE 11

Tietokone tallentaa tietoja

ABC80

Tieto DATA-lauseissa

Ohjelmassa tarvittava tieto tallennetaan useimmiten DATA-lauseisiin. Tällainen tieto luetaan käskyllä READ. Jos ohjelmaa käytetään eri yhteyksissä niin, että joka ajossa käytetään eri tietoja, voidaan eri ohjelmat tallentaa sellaisinaan kasetille.

Jos ohjelma on laaja, on epäkäytännöllistä tallentaa koko ohjelma nauhalle joka kerta, kun sitä käytetään samantapaisissa tehtävissä. On riittävää tallentaa ainoastaan ne ohjelmarivit, joita muutetaan eri ajojen välillä. Tarkastelkaamme seuraavaa esimerkkiä:

```
10 PRINT CHR$(12)
20 READ A$
30 PRINT "HINNASTO",A$
40 PRINT
50 FOR K=1 TO 3 : READ P(K) : NEXT K
60 PRINT "NASKALI",P(1)
70 PRINT "LAPIO",P(2)
80 PRINT "TRAKTORI",P(3)
90 REM **********************
100 DATA "mk brutto"
110 DATA 20,150,93000
```

Tämä ohjelma kirjoittaa hinnaston suppeahkolle tuotevalikoimalle. Koeaja!

Aseta nauhuri nauhoitusasentoon ja tallenna ohjelma kasetille käskyllä

```
SAVE HINNASTO
```

Kun halutaan kirjoittaa saman tuotevalikoiman nettohinnasto, on riittävää muuttaa rivien 100 ja 110 DATA-lauseet. Poista edellinen ohjelma käskyllä NEW ja syötä näppäimistöllä uudet DATA-lauseet. Ne ovat

```
100 DATA"mknetto"
110 DATA 18,130,90000
```

Tallenna nämä DATA-lauseet kasetille välittömästi ohjelman HINNASTO perään käskyllä

```
SAVE NETTO
```

Kirjoita nyt jälleen uusi sisältö seuraaville DATA-lauseille:

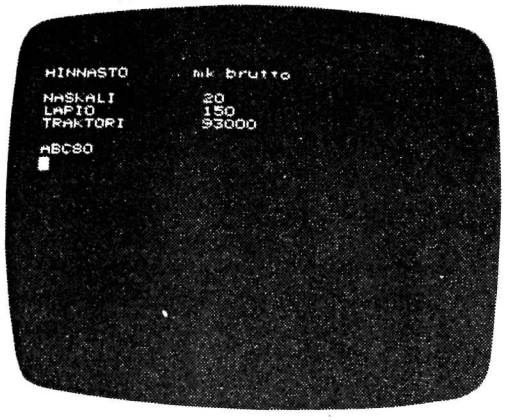
```
100 DATA Jälleenmyyjähinta
110 DATA 17,120,89000
```

Tallenna nämä rivit kasetille käskyllä

```
SAVE JÄLLEENM
```

Nyt voidaan kaikki kolme eri hinnastoa kirjoittaa näyttöön. Kelaa nauhuri alkuun, aseta se toistoasentoon ja kirjoita ensimmäinen hinnasto näyttöön käskyllä

```
RUN HINNASTO
```

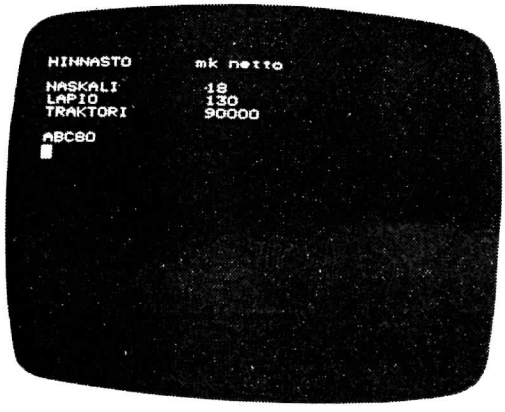


Bruttohinnasto

Kun nettohinnasto esitetään, täytyy ensin vaihtaa ohjelman DATA-lauseet. Uudet DATA-lauseet siirretään nauhalta toimintakäskyllä

```
MERGE NETTO
```

Aja tavanomaisesti käskyllä RUN.

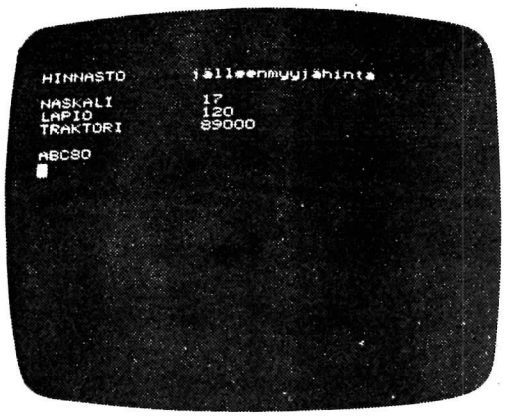


Nettohinnasto

Siirrä sitten jälleenmyyjien hinnat nauhalta käskyllä

```
MERGE JÄLLEENM
```

Aja tavalliseen tapaan.



Jälleenmyyjähinnasto

Laajenna ohjelmaa niin, että voidaan tallentaa suuremmankin tavaravalikoiman hinnasto. Muista varata tilaa DIM-lauseella, jos artikkeleiden määrä ylittää 10.

Tiedostot säästävät tilaa

Vaihtoehtona tiedon tallentamiselle DATA-lauseisiin on mahdollisuus käyttää ns. tiedostoja. Tällöin luodaan kasetille tilaa, johon voidaan kirjoittaa tietoja lähes samoin kuin näyttöruutuun. Tämän jälkeen nämä tiedot voidaan hakea nauhalta suurin piirtein samoin kuin ne kirjoitettaisiin näppäimistöllä.

Käskyä PREPARE . . .ASFILE . . . käytetään tällaisen tilan luomiseen nauhalle. Se nimi, joka halutaan antaa tiedostolle, sijoitetaan sanojen PREPARE ja ASFILE väliin. Nimi on ympäröitävä lainausmerkeillä. Tiedostolle on myös annettava numero, joka kirjoitetaan käskyn ASFILE jälkeen.

Esimerkiksi

```
10 PREPARE "DATA" ASFILE 1
```

Kun rivi 10 suoritetaan, luodaan kasetille tiedosto, jonka nimi on DATA ja numero on 1. Tietokone käynnistää nauhurin ja kirjoittaa tiedoston nimen nauhalle. Tiedon kirjoittamiseksi tiedostoon käytät käskyä PRINT lisättynä numeromerkillä (**#**) sekä tiedoston numerolla ja pilkulla, esim. PRINT **#** 1, . Muista, että merkki **#** aiheuttaa kirjoittamisen tiedostoon, ei kuvaruutuun.

Esimerkiksi

```
20 PRINT#1,"BRUTTO"
```

Jos haluat kirjoittaa useita tietoja samaan tiedostoon, lisää niitä vastaavat lauseet, esim.

```
30 PRINT#1,"20"
40 PRINT#1,"150"
50 PRINT#1,"93000"
```

Jos tiedostoon ei haluta sijoittaa enempää tietoja, täytyy tiedoston loppu merkitä ja nauhuri pysäyttää. Sen teet käskyllä CLOSE, jonka perään kirjoitat tiedoston numeron, esim.

```
60 CLOSE 1
```

Täydellinen ohjelma, joka luo tilan tiedostolle ja kirjoittaa siihen joitain tietoja, näyttää seuraavalta:

```
10 PREPARE "DATA" ASFILE 1
20 PRINT #1,"BRUTTO"
30 PRINT #1,"20"
40 PRINT #1,"150"
50 PRINT #1,"93000"
60 CLOSE 1
```

Tieto siirretään nauhalle

Aseta nauhurin kytkimet nauhoitusta varten, nollaa laskuri ja aja ohjelma. Kun tietokone on luonut tiedoston ja kirjoittanut sillä olevat tiedot sekä päättänyt tiedoston, näyttöön ilmestyy kuten tavallista ABC 80.

Käsky, joka valmistaa tiedostosta lukemisen, on OPEN . . . ASFILE . . ., jossa sanojen OPEN ja ASFILE väliin kirjoitetaan sitaatteihin halutun tiedoston nimi. Sanan ASFILE perään merkitään tiedoston numero.

Esimerkiksi

```
15 OPEN "DATA" ASFILE 1
```

Kun rivi 15 suoritetaan, nauhuri käynnistyy automaattisesti ja tietokone alkaa etsiä tiedostoa, jonka nimi on DATA. Tiedon lukemiseksi tiedostosta käytetään käskyä INPUT lisättynä numeromerkillä, tiedostonumerolla ja pilkulla, esim. INPUT # 1, .

Kaksi esimerkkiä:

```
21 INPUT #1,A$
25 INPUT #1,P(1),P(2),P(3)
```

Lopettaaksesi tiedoston lukemisen ja pysäyttääksesi nauhurin käytä käskyä CLOSE ja tiedostonumeroa, esim.

```
27 CLOSE 1
```

Tarkistaaksesi, että halutut tiedot ovat siirtyneet nauhalta lisäät rivin

```
28 PRINT A$,P(1),P(2),P(3)
```

Täydellinen ohjelma, joka lukee tietoja tiedostosta on siten

```
15 OPEN "DATA" ASFILE 1
21 INPUT #1,A$
25 INPUT #1,P(1),P(2),P(3)
27 CLOSE 1
28 PRINT A$;P(1);P(2);P(3)
```

Koeaja ohjelma! Älä unohda, että sitä ennen on nauhan oltava takaisinkelattuna ja nauhurin valmiina toistoon.

Yhdistä tämä ohjelma luvun alussa olevaan ohjelmaan HINNASTO kelaamalla nauha takaisin ja antamalla käsky

```
MERGE HINNASTO
```

Poista nyt data-rivit 90, 100 ja 110 sekä rivit 20, 28 ja 50. Kun käytät käskyä LIST on tuloksena seuraava ohjelma:

```
10 PRINT CHR$(12)
15 OPEN "DATA" ASFILE 1
21 INPUT #1,A$
25 INPUT #1,P(1),P(2),P(3)
27 CLOSE 1
30 PRINT "HINNASTO",A$
40 PRINT
60 PRINT "NASKALI",P(1)
70 PRINT "LAPIO",P(2)
80 PRINT "TRAKTORI",P(3)
```

Ohjelmaa ajettaessa tietokone etsii tiedoston DATA ja kirjoittaa sitten näyttöön bruttohinnaston samoin kuin luvun ensimmäisessä esimerkissä.

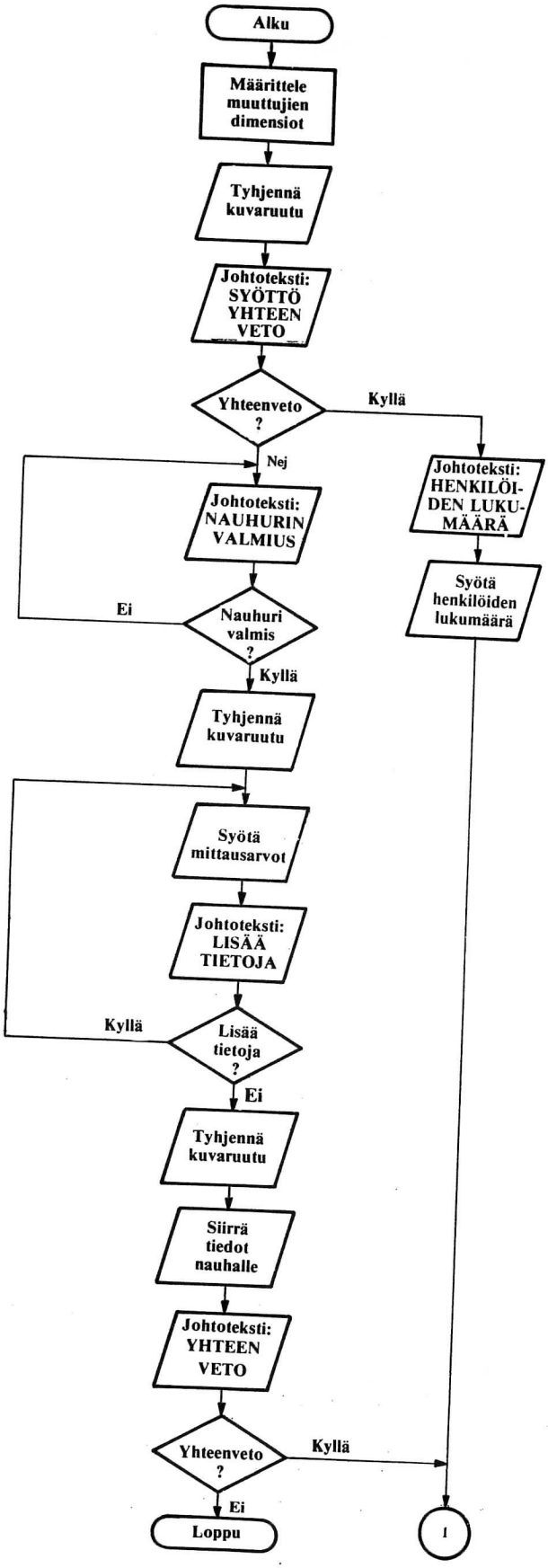
Käyttäjälle ystävällinen ohjelma

Tässä esimerkissä, jota ei ole tarkoitettu otettavaksi liian vakavasti, rekisteröi rykmentin lääkäri mittaustuloksensa, jotka ilmoittavat asevelvollisten pituuden ja vyötärön ympärysmitan. Näitä tärkeitä mittaustuloksia voivat tietenkin käyttää hyväkseen myös muut asiasta kiinnostuneet, kuten varustevaraston hoitaja, joka pitää huolen siitä, että asevelvolliset saavat käyttökelpoiset vaatteet. Keittiössä voidaan näiden tietojen avulla laskea ruuankulutus tarkasti jne.

Tutki vuokaaviota seuraavalla aukeamalla ja koeta konstruoida ne ohjelmanosat, jotka siihen sisältyvät. Valmiissa ohjelmassa on eri ohjelmanosat erotettu toisistaan välein helpottamaan vuokaavion ja valmiin ohjelman välistä vertailua. Vuokaavion symboleihin voidaan kirjoittaa ne ohjelmarivien numerot, joilla kyseinen osa suoritetaan.

```
10 DIM L(100) : DIM M(100)

20 PRINT CHR$(12)

30 PRINT "ILMOITA , ONKO KYSEESSÄ"
31 PRINT "-TIETOJEN TALTIOINTI
NAUHALLE <T>"
32 PRINT "-TIETOJEN LUKU NAUHALTA <L>"

40 GET A$ : IF A$="L" THEN GOTO 330

50 PRINT
60 PRINT "HUOM !ONKO NAUHURI VALMIS
<K> ?";

70 GET A$
80 IF A$="K" THEN GOTO 90 ELSE GOTO 60

90 PRINT CHR$(12)

100 PRINT : PRINT
110 I=I+1
120 PRINT "ILMOITA MITATTU PITUUS
(cm)";
130 INPUT P(I)
140 PRINT "MITATTU VYÖTÄRÖNLEVEYS
(cm)";
150 INPUT V(I)
160 PRINT : PRINT
170 PRINT "TULEEKO LISÄÄ TIETOJA
( K tai E )"
```

```
180 GET A$
190 IF A$="K" THEN GOTO 90

200 PRINT CHR$(12)

210 PRINT "TIEDOT SIIRRETÄÄN
KASETILLE !"
220 PREPARE "SAINT" ASFILE 1
230 FOR X=1 TO I
240 PRINT #1,P(X)
250 PRINT #1,V(X)
260 NEXT X
270 CLOSE 1
280 PRINT : PRINT "SIIRTO ON SELVÄ"

290 PRINT "HALUTAANKO YHTEENVETO
(K tai E)";

300 GET A$
310 PRINT
320 IF A$="K" THEN GOTO 350 ELSE END

330 PRINT "KUINKA MONTA HENKEÄ ";

340 INPUT I

350 PRINT "ONKO NAUHA KELATTU TAKAISIN
(K)";

360 GET A$
370 PRINT
380 IF A$="K" THEN GOTO 390 ELSE 350

390 PRINT
400 PRINT "ONKO NAUHURI VALMIS (K)";

410 GET A$
420 PRINT
430 IF A$="K" THEN GOTO 440 ELSE 400

440 PRINT
450 PRINT "PITUUS","VYöTÄRöNLEVEYS"
460 PRINT
470 OPEN "SAINT" ASFILE 1
480 FOR T=1 TO I : INPUT #1,P(T)
490 INPUT #1,V(T)
500 PRINT P(T),V(T)
510 NEXT T : CLOSE 1

520 END
```


Alku
Määrittele muuttujien dimensiot
Tyhjennä kuvaruutu
Johtoteksti: SYÖTTÖ YHTEEN VETO
Yhteenveto ?
Kyllä
Nej
Johtoteksti: NAUHURIN VALMIUS
Johtoteksti: HENKILÖIDEN LUKUMÄÄRÄ
Syötä henkilöiden lukumäärä
Ei
Nauhuri valmis ?
Kyllä
Tyhjennä kuvaruutu
Syötä mittausarvot
Johtoteksti: LISÄÄ TIETOJA
Kyllä
Lisää tietoja ?
Ei
Tyhjennä kuvaruutu
Siirrä tiedot nauhalle
Johtoteksti: YHTEEN VETO
Yhteenveto ?
Kyllä
Ei
Loppu
1
1
Johtoteksti: NAUHA KELATTU
Ei
Selvä ?
Kyllä
Johtoteksti: NAUHURI VALMIS?
Ei
Selvä ?
Kyllä
Tieto siirtyy nauhalta
Loppu

Yhteenveto

Käsky	Selitys	Esimerkki
PREPARE... ASFILE...	Luo uuden tiedoston, jolla on tietty numero väliltä 1—255. Nauhuri käynnistyy.	10 PREPARE "KIRJE1.TXT" ASFILE 2
OPEN... ASFILE...	Sijoittaa käytössä olevalle tiedostolle tietyn numeron väliltä 1—255. Nauhuri käynnistyy.	20 OPEN "KIRJE1.TXT" ASFILE 2
CLOSE	Sulkee tiedoston eli lopettaa tiedoston lukemisen tai siihen kirjoittamisen. Nauhuri pysähtyy.	50 CLOSE 2
PRINT#...;	Kirjoittaa tietoja nauhalla olevaan tiedostoon.	30 PRINT#1;X
INPUT#...;	Syöttää tietoja nauhalla olevalta tiedostolta.	40 INPUT#1;Y
INPUTLINE#...;	Lukee yhden rivin nauhalla olevalta tiedostolta.	50 INPUTLINE# 1;A$

Tiedostonimen pituus voi olla aina kahdeksan merkkiä, lisättynä mahdollisesti jollain merkinnällä kuten .TXT, jota käytetään tiedostojen yhteydessä.

Harjoituksia

11.1 Kirjoita ohjelma, joka lukee 20 näppäimistöllä syötettyä lukua muistiin ja sitten tallettaa nämä arvot tiedostoon.

11.2 Tee ohjelma, joka lukee edellisessä tehtävässä talletetut 20 lukua tiedostosta ja tulostaa ne kuvaruutuun.

KAPPALE 12

Tietokone laskee merkkimuuttujia

ABC80

Kokonaislukumuuttujat

Emme ole tähän mennessä tutkineet tarkemmin sitä eroa, joka on luvun esittämisessä joko kokonaisluku- tai liukulukumuodossa. Tarkastelkaamme esimerkkiä, jossa ero tulee selvästi esille.

```
10 FOR X=1 TO 10000 : NEXT X
20 PRINT "SELVÄ"
```

Aja yllä oleva ohjelma ja mittaa aika, joka kuluu ajoon. Se on noin 10 sekuntia. Muuta sen jälkeen ohjelmaa siten, että silmukkamuuttujaksi tulee kokonaislukumuuttuja, ja mittaa ajoaika uudestaan.

```
10 FOR X%=1% TO 10000% : NEXT X%
20 PRINT "SELVÄ"
```

Tässä tapauksessa tulee ajoajaksi noin kaksi sekuntia. Ohjelmanajo tapahtui siis huomattavasti nopeammin, kun käytimme kokonaislukumuuttujaa silmukkamuuttujana. Puuttumatta lähemmin puhtaasti teknisiin selityksiin toteamme, että ohjelma suoritetaan nopeammin, kun käytetään kokonaislukumuuttujaa. Mikäli käytettävissä oleva lukualue (—32768—32767) sallii, tulee siis käyttää kokonaislukumuuttujia liukulukumuuttujien asemesta. Riville 10 sisältyy myös kokonaislukuvakioita. Koska kokonaislukuvakiot käyttävät vain puolet muistitilasta verrattuna liukulukuvakioiden tarvitsemaan tilaan, tulee ensinmainittuja käyttää mahdollisuuksien mukaan tietokoneen muistitilan säästämiseksi.

Liukulukumuuttujat

Liukulukumuuttujien lukualue (±0. 1E-127—±0.999999E+127) on riittävä useimpiin sovelluksiin. Sitä vastoin voidaan, kuten olemme nähneet, niin pienet kuin suuretkin luvut ilmaista enintään kuuden numeron tarkkuudella. Monissa laskuissa tämä ei ole riittävää. Tietyissä sovelluksissa, kuten liikevaihtoverolaskuissa, on toisaalta vaivalloista lukea liian monidesimaalista näyttöä. ABC 80 tarjoaa kuitenkin mahdollisuuden käyttää laskutoimituksissa numeerisia jonoja (numerojonoja) mainittujen liukulukumuuttujien käyttöä koskevien puutteiden korjaamiseksi.

Numerojonot

ABC 80 voi laskea suurella tarkkuudella (aina 29 numeroon saakka) ns. ASCII-aritmetiikan avulla. ASCII-aritmetiikka käyttää hyväkseen jonoja, jotka voidaan tulkita lukuina. Numerojono voi sisältää merkit +, —, numeroita ja desimaalipisteen. Sitävastoin numerojonoon ei voi sisältyä eksponenttisymbolia E. Eksponenttimuoto ei siten ole sallittu.

Esimerkkejä numerojonoista:

```
.123456789
2003.456
-435.39421076
```

Numeerisille jonomuuttujille annetaan nimi kuten tavallisille jonomuuttujille ja niitä käsitellään samalla tavalla. Koska nyt syötetään pelkkiä numeroita, ei jonoja tarvitse ympäröidä sitaatein.

ASCII-aritmetiikka

Numerojonojen yhteydessä käytettävät laskutoimituskäskyt ovat ADD¤(A¤, B¤, N), SUB¤(A¤, B¤, N), MUL¤(A¤, B¤, N) ja DIV¤(A¤, B¤, N). Ne tulkitsevat jonot A¤ ja B¤ lukuina ja suorittavat niiden välisen yhteen-, vähennys-, kerto- ja jakolaskun. Tulos pyöristetään N:n desimaalin tarkkuuteen.

Käsky COMP%(A¤, B¤) vertaa numerojonoja A¤ ja B¤ keskenään lukuina. Tulos on —1, jos luku A¤ on pienempi kuin B¤, 0, jos A¤ = B¤, ja 1, jos luku A¤ on suurempi kuin B¤.

Seuraavaa ohjelmaa voidaan käyttää neljän peruslaskutoimituksen suorittamiseen ASCII-aritmetiikalla. Käytämme ABC 80:n pikku hienoutta merkitä puolipiste käskyn PRINT lyhenteenä.

```
10 ; CHR¤(12)
20 ; "YHTEENLASKU     (1)"
30 ; "VÄHENNYSLASKU   (2)"
40 ; "KERTOLASKU      (3)"
50 ; "JAKOLASKU       (4)"
60 ;
70 ; "MITÄ LASKUTAPAA KÄYTETÄÄN ";
80 INPUT I
90 ;
100 ; "ANNA LUVUT A JA B ";
110 INPUT A¤,B¤
120 ;
130 ; "MONELLAKO DESIMAALILLA VASTAUS ";
140 INPUT N
```

```
150 ;
160 ON I GOSUB 180,190,200,210
170 GOTO 60
180 ; "SUMMA=";ADD¤(A¤,B¤,N) : RETURN
190 ; "EROTUS=";SUB¤(A¤,B¤,N) : RETURN
200 ; "TULO=";MUL¤(A¤,B¤,N) : RETURN
210 ; "OSAMÄÄRÄ=";DIV¤(A¤,B¤,N) :
RETURN
```

Numerojonojen välinen laskutoimitus valitaan INPUT-lauseella (rivi 80), missä 1 = yhteenlasku, 2 = vähennyslasku, 3 = kertolasku ja 4 = jakolasku.

Numerojonoille A ¤ ja B ¤ sijoitetaan arvot seuraavalla INPUT-lauseella. Rivillä 160 on uusi käsky, jota et ole tavannut aikaisemmin. Käsky ON I GOSUB 200, 300, 400, 500 saa aikaan hypyn tietylle ohjelmariville, josta haluttu laskutoimitus saadaan suoritetuksi. Muuttujan I arvo määrää, mille riville hyppy tapahtuu.

ASCII-aritmetiikalla suoritetun laskutoimituksen tulos ei voi ylittää 29:ää numeroa. Laskutoimitusten tarkkuudessa esiintyy vaihteluita, mutta normaalitapauksissa se on enemmän kuin 20 numeroa.

ON-lauseet

Lausetta ON . . . GOSUB . . . käytetään, kun halutaan hypätä ohjelmassa eri aliohjelmiin riippuen, siitä mikä arvo muuttujalla tai lausekkeella on.

ON I GOSUB 200, 300, 400 saa aikaan hypyn riville 200, jos I = 1, riville 300, jos I = 2, ja riville 400, jos I = 3. Kaikki muut I:n arvot antavat virheilmoituksen.

Muut ABC 80:ssa olevat ON-lauseet ovat ON . . . GOTO . . ., ON . . . RESTORE . . . ja ON ERROR GOTO . . .

Lausetta ON . . .GOTO . . . käytetään, kun halutaan hypätä useaan eri paikkaan ohjelmassa muuttujasta riippuen. Se toimii kuten lause ON . . . GOSUB . . .

RESTORE merkitsee sitä, että seuraavalla READ-lauseella palataan ensimmäiseen DATA-lauseeseen tiedon hakemiseksi. Lausetta ON ... RESTORE ... käytetään, kun halutaan hakea tietoja täysin määrätyistä DATA-lauseista.

ON I RESTORE 200, 300, 400 sisältää sen, että tietokone alkaa hakea DATA-lausetta riviltä 200 alkaen, jos I = 1, riviltä 300 alkaen, jos I = 2, jne.

ON ERROR GOTO ... määrää hypyn tietylle riville, jos ohjelmanajon aikana tapahtuu virhe. Tämän käskyn käytöstä opit lisää 13. luvussa.

Oikea pyöristäminen

Kun käytät ASCII-aritmetiikkaa, tulos saadaan aina oikein pyöristettynä. Kirjoita edellä annettu ohjelma koneeseen ja käytä sitä jakolaskun 2/3 laskemiseksi 25 desimaalin tarkkuudella. Ohjelmanajo antaa seuraavan näytön:

Osamäärä 2/3 laskettuna 25 desimaalilla

Hyvin suuria lukuja

Jos kokonaislukumuuttujan lukualue on liian pieni tai liukulukumuuttujaan liittyvä kuuden numeron tarkkuus ei riitä, on paikallaan käyttää ASCII-aritmetiikkaa. Käyttäkäämme sitä potenssin x^N laskemiseksi, erityisesti suurilla N:n arvoilla.

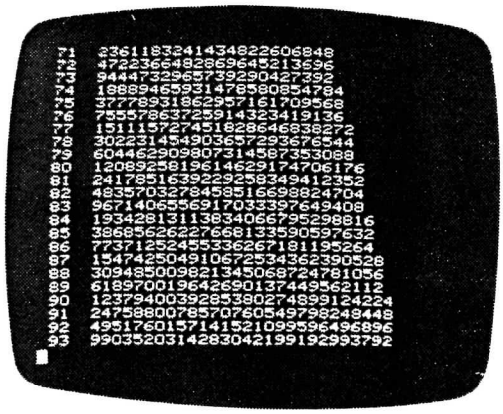
Seuraava ohjelma laskee potenssin x^N, jossa x on kokonaisluku väliltä 1—9.

```
10 ; CHR¤(12)
20 ; "POTENSSIINKOROTUS X**N"
25 ;
30 ; "Jokainen arvon X painallus
(esim. 2)"
40 ; "lisää yhden yksikön N:ään ja "
50 ; "N ja X**N tulostuvat kuvaruutuun."
60 GET A¤
70 B¤=A¤
80 N=1
90 N=N+1
100 A¤=MUL¤(A¤,B¤,0)
110 IF N>2 THEN 150
120 ;
130 ; " N";TAB(5);"X**N"
140 ;
150 ; N;TAB(5);A¤
160 GET C¤
170 IF C¤=B¤ THEN 90
```

Kantaluvun x arvo syötetään GET-lauseella numerojonolle A¤. Kun näppäintä x painetaan, kasvaa N yhdellä ja uusi arvo lasketaan. Jos pidät näppäintä x koko ajan painettuna, ajo sujuu nopeammin, koska kaikki näppäimet ovat jatkuvatoimisia.

Kirjoita ohjelma ja aja se muutamilla x:n eri arvoilla!

Seuraava kuva näyttää ohjelman tulostuksen, kun x = 2. Vain viimeisimmät tulokset ovat näytössä. Rajanahan on 29 numeroa.



Potenssin x^N laskeminen ASCII-aritmetiikalla

Haluttu desimaalien lukumäärä

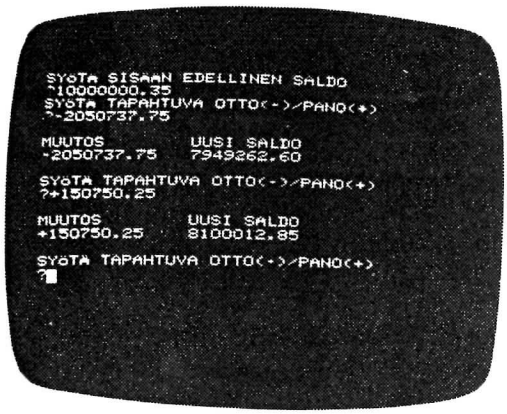
ASCII-aritmetiikka antaa mahdollisuuden valita desimaalien lukumäärä laskutehtävän vaatimuksiin sopivana, esim. kaksi desimaalia rahamääriin liittyvissä laskuissa.

Seuraava ohjelma käyttää ASCII-aritmetiikkaa pankkitilitapahtumien laskemiseksi. Tässä tutusssa ohjelmassa on nyt oleellista muuttujien muuttaminen jonomuotoon.

```
5 PRINT CHR$(12)
10 PRINT
20 PRINT "SYÖTÄ SISÄÄN EDELLINEN SALDO",
30 INPUT A$
40 PRINT "SYÖTÄ TAPAHTUVA OTTO(-)/PANO(+)"
50 INPUT B$
60 PRINT
70 A$=ADD$(A$,B$,2)
80 PRINT "MUUTOS","UUSI SALDO"
90 PRINT B$,A$
100 PRINT
110 GOTO 40
```

Monimiljonääri Riklund haluaa käyttää ABC 80 tietokonettaan laskeakseen pankkitilinsä saldon ostettuaan ensin huvipurren, jonka hinta oli 2 050 737,75 mk. Samana päivänä hän oli saanut osasuorituksen mk 150 750,25 konemyynnistään. Hänen pankkitilinsä edellinen saldo oli mk 10.000 000,35.
Kirjoita ohjelma ja aja se edellä esitetyin luvuin.

Tulostus näyttää seuraavalta:



Tiliote ASCII-aritmetiikalla

Yhteenveto

Käsky	Selitys	Esimerkki
RESTORE	Määrää seuraavan READ-lauseen lukemaan tietoja tietyiltä DATA-lauseilta. Eri vaihtoehtoja ovat:	
	Lukee tietoja ensimmäisestä DATA-lauseesta alkaen.	`10 RESTORE`
	Lukee tietoja riviltä 200 alkaen.	`10 RESTORE 200`
ON... GOTO...	Hyppää eri ohjelmariveille muuttujan arvon perusteella. Esimerkissä antaa arvo N=1 hypyn riville 200, arvo N=2 hypyn riville 260 jne.	`10 ON N GOTO 200, 260, 310`

ON... GOSUB...	Hyppää eri aliohjelmiin muuttujan arvon perusteella. Esimerkissä antaa arvo N=1 hypyn riville 30, arvo N=2 hypyn riville 120 jne.	`10 ON N GOSUB 30, 120, 400`
ON... RESTORE...	Määrää RESTORE käskyn eri DATA-lauseisiin muuttujan arvon perusteella. Esimerkissä luetaan arvolla N=1 tietoa riviltä 60 alkaen, arvolla N=2 riviltä 70 alkaen jne.	`10 ON N RESTORE 60,70,80`
ON ERROR GOTO	Määrää hypyn tietylle riville, jos ohjelmanajossa tulee virhe. Hyppy voidaan liittää vain tähdellä merkittyihin virhekoodeihin (liite 2). Esimerkissä aiheuttaa virheilmoitus hypyn riville 500.	`10 ON ERROR GOTO 500`
;	Käskyn PRINT lyhenne.	`10 ; "HEI"`

Jonofunktio	Selitys
ADD¤(A¤,B¤,N)	Laskee yhteen jonot A¤ ja B¤ lukuina. Tulos pyöristetään N:n desimaalin tarkkuuteen. Jonot voivat sisältää enintään 29 numeroa sekä etumerkin + tai — ja desimaalipisteen mutta ei eksponenttia.

SUB¤(A¤,B¤,N)	Vähennyslasku (kuten edellä). B¤ vähennetään A¤:sta ja tulos pyöristetään N desimaalin tarkkuuteen.
MUL¤(A¤,B¤,N)	Kertolasku. A¤ ja B¤ kerrotaan keskenään.
DIV¤(A¤,B¤,N)	Jakolasku. A¤ jaetaan luvulla B¤.
COMP%(A¤,B¤)	Vertailee lukuja A¤ ja B¤. Vertaa kahta numerojonoa keskenään. Antaa tuloksen: —1 jos A¤ < B¤ 0 jos A¤ = B¤ 1 jos A¤ > B¤

Harjoituksia

12.1 Kirjoita ohjelma, joka laskee kiinteistönvälittäjän neljän prosentin välityspalkkion, ja laske seuraavista kiinteistökaupoista saatu palkkio:

Myyntihinta: Talo A 5.240.733,—
Talo B 1.750.116,—
Talo C 3.777.750,—
Talo D 2.125.388,—

Kirjoita tulokset kahden desimaalin tarkkuudella taulukkomuotoon.

KAPPALE 13

Tietokone etsii ohjelmointivirheitä

ABC80

Suunnittelu

Useimmat ohjelman virheet (ohjelmointivirheet!) voidaan välttää määrittelemällä tietokoneen suoritettavaksi annettava tehtävä tarkasti ennen ohjelman kirjoittamista. Ohjelmahan on varsinaisesti vain hyvin yksityiskohtainen kuvaus tietokoneen tehtävästä. Olemme jo aikaisemmin, luvussa 7, kuvanneet etuja, joita saavutetaan käyttämällä vuokaaviota ensiaskeleena ongelman käsittelyssä.

On edullista käyttää aliohjelmia tavallisten hyppykäskyjen asemesta, missä se vain on mahdollista. Ohjelman rakentaminen helpottuu huomattavasti ja havainnollisuus paranee.

Mitä enemmän ohjelmaan on sisällytetty tiedonantoja REM- ja PRINT-lauseilla, sitä helpompaa on ymmärtää ohjelmaa.

Tietokoneen ilmoitukset virheistä ovat suuri apu, kun korjataan virheellisiä käskyjä, tiedostoja ym.

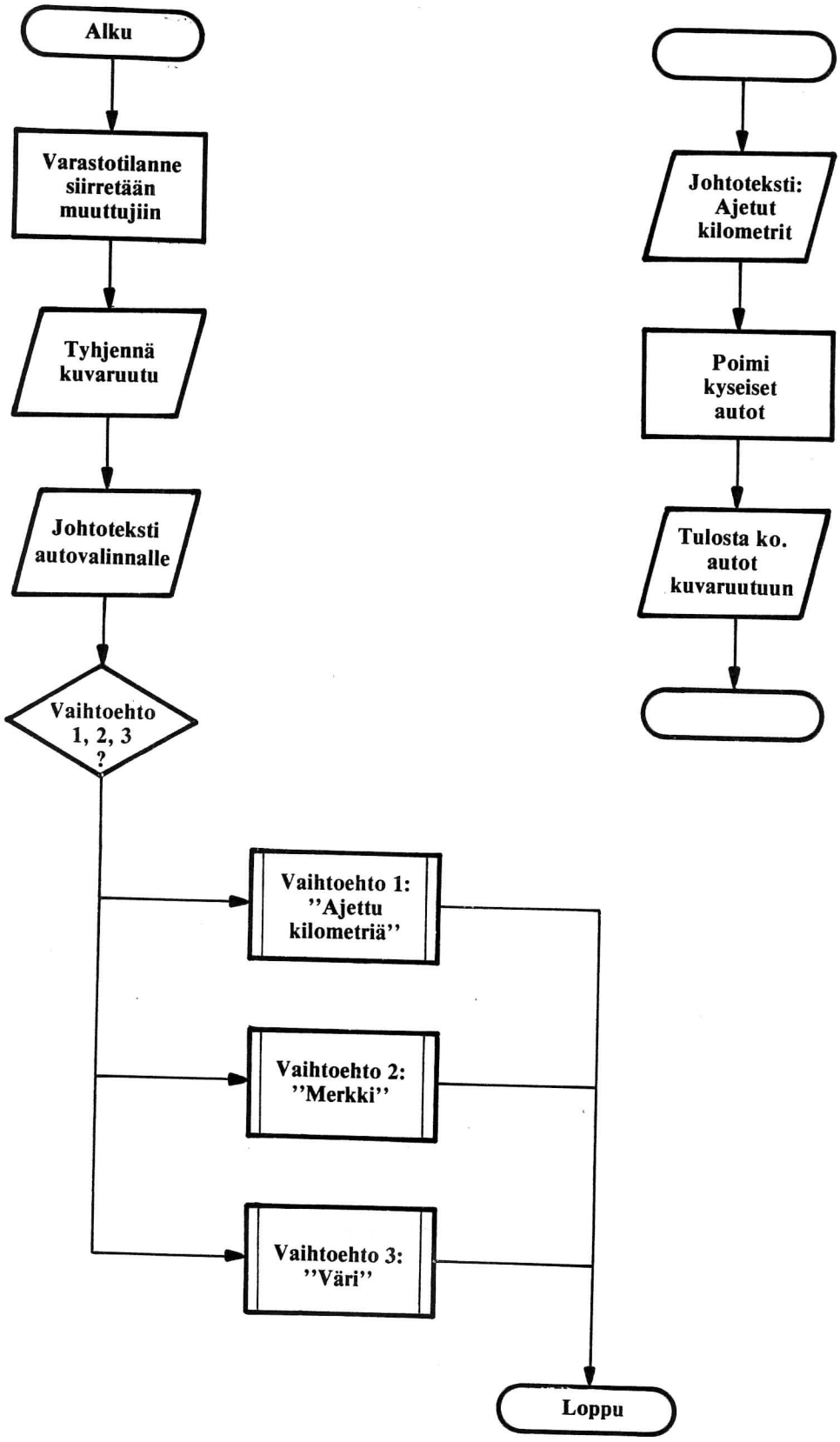
Ohjelmaesimerkki

Lähtökohtana seuraavalle keskustelulle käytämme ohjelmaesimerkkiä, johon on soluttautunut yksi jos toinenkin virhe. Annetaan tietokoneen tehtäväksi esitellä autokaupan varastosta tietyt autot asiakkaan vaatimusten ja toivomusten mukaisesti.

Jos asiakas on kiinnostunut noin 70 000 km ajetusta autosta, hänen ei siis tarvitse lukea läpi pitkää listaa kaikista myytävänä olevista autoista, vaan hän voi kysyä tietokoneelta, mitkä autot sopivat hänen toiveisiinsa.

Jos ostaja edellyttää tiettyä automerkkiä tai erityistä väriä, tulee tietokoneen osata valita myös nämä toiveet täyttävä auto. Kahden myöhemmän toivomuksen täyttävät ohjelmanosan saat laatia itse. (Harjoitustehtävät)

Ohjelmaesimerkin vuokaavio on karkeasti hahmoteltuna seuraavalla sivulla. Ratkaisu rakentuu lyhyeen johdantoon, jota seuraa kolme vaihtoehtoista aliohjelmaa. Niistä on vain ensimmäinen kirjoitettu vuokaaviona. Kaksi muuta aliohjelmaa korvataan tässä vaiheessa vain lyhyellä ilmoituksella valitusta vaihtoehdosta ja tiedotuksella, ettei ohjelma ole näiltä osin vielä valmis.


Alku
Varastotilanne siirretään muuttujiin
Tyhjennä kuvaruutu
Johtoteksti autovalinnalle
Vaihtoehto 1, 2, 3 ?
Vaihtoehto 1: ''Ajettu kilometriä''
Vaihtoehto 2: ''Merkki''
Vaihtoehto 3: ''Väri''
Loppu
Johtoteksti: Ajetut kilometrit
Poimi kyseiset autot
Tulosta ko. autot kuvaruutuun

Vuokaavion avulla kirjoitettu ohjelma, joka siis sisältää joitain virheitä, on seuraava:

```
10 FOR K=1 TO 6
20 READ A$,B$,C
40 PRINT CHR$(12)
50 PRINT "MINKÄ OMINAISUUDEN MUKAAN
HALUAT VALITA AUTOSI ?:"
60 GOSUB 120
70 PRINT "1. AJETTU km-MÄÄRÄ" : PRINT
"2. MERKKI"
80 PRINT "3. VÄRI"
90 INPUT A
100 ON A GOSUB 140,210,230
110 END
```

Loput ohjelmasta koostuu aliohjelmista:

```
120 PRINT "(VASTAA ANTAMALLA KYSEINEN
LUKU)"
130 PRINT : RETURN
140 PRINT "ILMOITA km-MÄÄRÄ (noin ! )"
150 GOSUB 120
160 INPUT A
170 FOR K=1 TO 6
180 IF ABS(A-C(K))>25000 THEN 200
190 PRINT A$(K),B$(K),C(K)
200 NEXT K : RETURN

210 PRINT "KOHTAA 2. KOSKEVA OHJELMAN
OSA EI VIELÄ OLE VALMIS" : RETURN
220 PRINT "KOHTAA 3. KOSKEVA OHJELMAN
OSA EI VIELÄ OLE VALMIS" : RETURN

230 DATA VOLKSWAGEN,SININEN,40000
240 DATA OPEL,VIHREÄ,50000
250 DATA FIAT,KELTAINEN,20000
260 DATA VOLVO,PUNAINEN,70000
270 DATA AUDI,VALKOINEN,80000
280 DATA SAAB,RUSKEA,30000
```

Tietokone antaa virheilmoituksen

Kun nyt ajat ilmeisen virheellisen ohjelmaesimerkin, tietokone antaa ilmoituksen

```
ERR 24 LINE 10
```

Tämä sisältää sen, että käskyn NEXT sisältävä rivi puuttuu. Lisäksi ilmoitetaan, että puuttuva silmukkamuuttuja löytyy riviltä 10.

```
10 FOR K=1 TO 6
```

Puuttuva käsky on siis

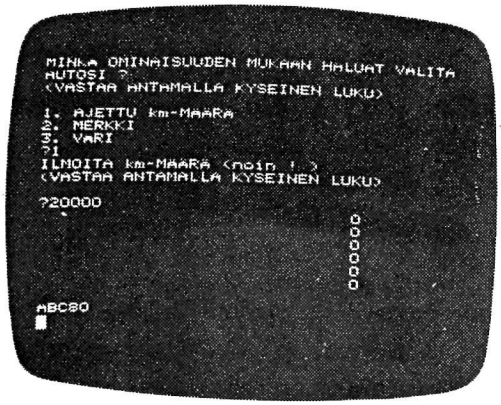
```
30 NEXT K
```

Kirjoita se ohjelmaan!

Ne virhetyypit, joiden takia tietokone ei tunne käskyä tai ei jostain syystä osaa suorittaa sitä, ovat helposti korjattavia. Liitteessä 1 on luettelo kaikista tietokoneen virheilmoituksista. Luvun lopussa on lisäksi selitetty, kuinka tietokone voi ilmoittaa virheen selväkielisenä.

Ohjelmavirheiden jäljittäminen toimintakäskyllä TRACE

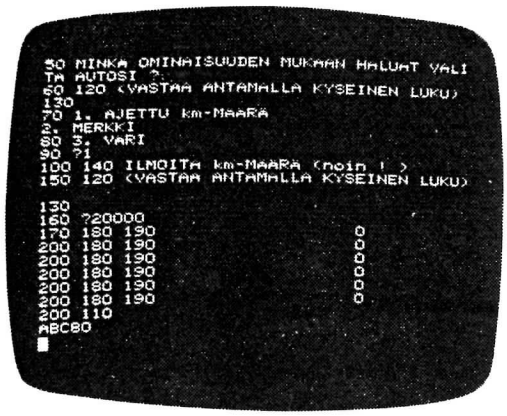
Aja ohjelma muutamia kertoja ja aloita vaihtoehdolla 1. Kun olet vastannut ajokilometrimäärään, tietokone antaa valitettavasti väärän tiedon. Ohjelman mukaan annetaan vastaus kolmessa sarakkeessa; merkki, väri ja km-lukema. Näyttöön saadaan kuitenkin vain kaksi tyhjää saraketta ja yksi sarake pelkkiä nollia. Mahdollista on saada myös näyttöön pelkkä ABC 80.



Vaihtoehdon 1 ajo

Komennon TRACE avulla voidaan tutkia, missä järjestyksessä käskyt suoritetaan.

Anna nyt komento TRACE ja aja uudestaan. Kun olet syöttänyt luvun 1 vaihtoehdoksi ja 20 000 kilometrilukemaksi, saadaan näytöksi



Ajo toimintakäskyllä TRACE

Näytön vasemmassa reunassa olevat luvut

```
50
60  120
130
70    jne.
```

ovat suoritettujen rivien numeroita.

Esimerkiksi rivi

```
60 GOSUB 120
```

määrää hypyn aliohjelmaan riveillä 120 ja 130. Aliohjelmasta hypätään takaisin riville 70.

Rivinumeroiden perässä on joskus tekstiä, esim.

```
50 MINKÄ OMINAISUUDEN MUKAAN HALUAT
VALITA AUTOSI ?:
60 120 (VASTAA ANTAMALLA KYSEINEN LUKU)
```

Tämän selitys on, että todellisilla ohjelmariveillä on PRINT-lauseita, esim.

```
120 PRINT "(VASTAA ANTAMALLA KYSEINEN
LUKU)"
```

Jos ohjelmassa ei ole PRINT-lauseita, kirjoitetaan näyttöön pelkät rivinumerot. Riveillä 170, 180, 190 ja 200 oleva silmukka suoritetaan kuusi kertaa kuten pitääkin. Rivillä 190 annettava näyttö on kuitenkin joka kerta nolla. Se on virhe, ja meidän täytyy siis tutkia, mistä se johtuu.

Toteamme samanaikaisesti toisen ongelman. Tässä yhteydessä ei TRACE anna tietoa tapahtumista riveillä 10—40. Se johtuu siitä, että monitori tyhjennetään käskyllä

```
40 PRINT CHR$(12)
```

Tästä vaikeudesta selviämme lisäämällä pysäytyskäskyn ohjelmaan.

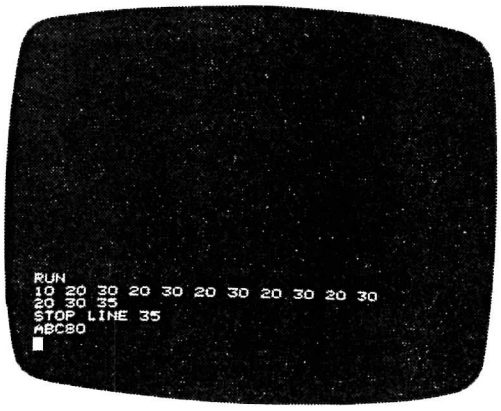
Tilapäisiä pysähdyksiä ohjelmassa

Usein voi siis olla käytännöllistä saada aikaan väliaikaisia pysähdyksiä ohjelmassa. Käsky STOP on erinomainen sitä varten. Käskyjen STOP ja END pääasiallinen ero on se, että STOP ei nollaa muuttujia.

Lisää seuraava rivi ja aja ohjelma

```
35 STOP
```

Koska käsky TRACE on jatkuvasti voimassa, saadaan seuraava näyttö



Ajon keskeytys käskyllä STOP

Nyt saadaan siis riveillä 10, 20, 30 ja 35 olevien käskyjen suoritusjärjestys selvitetyksi.

ABC 80 sallii monien käskyjen käyttämisen myös komentoina. Jos esimerkiksi kirjoitat PRINT-lauseen ilman rivinumeroa, PRINT tulkitaan komentona, ja haluttu tulostus saadaan välittömästi näppäimellä RETURN. Koska STOP ei nollaa muuttujia, voit antaa toimintakäskyn

```
PRINT A¤,B¤,C
```

Näyttöön saadaan

```
SAAB          RUSKEA          30000
```

Nämä ovat tiedot viimeisestä DATA-lauseiden autosta.

Käskyllä TRACE olemme todenneet aikaisemmin, että rivin 190 antama tulostus oli virheellinen. Rivi

```
190 PRINT A¤(K),B¤(K),C(K)
```

ei antanut mitään tulostusta ensimmäiseen ja toiseen sarakkeeseen, eli muuttujille A¤(K) ja B¤(K) ei ole annettu mitään arvoa. Kolmanteen sarakkeeseen saatiin pelkkiä nollia, eli myöskään C(K) ei ole saanut ohjelmassa mitään arvoa.

Sijoitus tapahtuu rivien 10, 20 ja 30 määräämällä silmukalla. Olemme nyt jäljittäneet niin paljon tietoa, että tiedämme virheen olevan rivillä 20. Siellä olisi tullut käyttää indeksoituja muuttujia!

Rivi kirjoitetaan siis seuraavassa muodossa

```
20 READ A¤(K),B¤(K),C(K)
```

Jotta ohjelma voitaisiin nyt ajaa normaalisti, on ensin annettava toimintakäsky NOTRACE. Huomaa, että käskyjä TRACE ja NOTRACE voidaan käyttää myös ohjelmassa. Poista myös rivi 35 STOP. Nyt ohjelma toimii, ainakin vaihtoehdon 1 osalta.

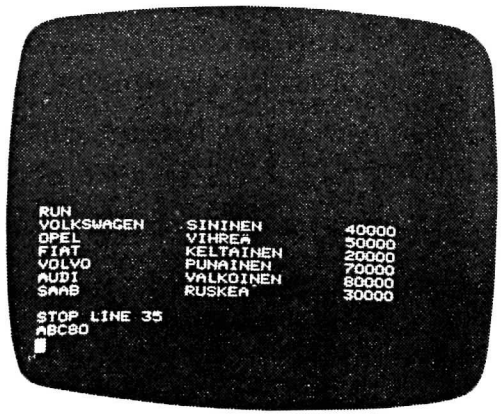
Ohjelman tutkiminen Print-lauseilla

Tässä tapauksessa voitiin virheet paljastaa melko helposti TRACE-käskyllä, koska riveillä 170—200 olevassa silmukassa on PRINT-lause. Jos tarvittava PRINT-lause ei ole ohjelman luonnollinen osa, on usein sopivaa lisätä tilapäisiä PRINT-lauseita muuttujanarvojen tarkastelemiseksi. Tämä koskee erityisesti silmukoita, joissa lisätty STOP aiheuttaa sen, että ohjelmaosa suoritetaan vain yhden kerran.

Lisää nyt PRINT-lause korjattuun silmukkaan riveillä 10, 20 ja 30 ja sijoita myös tilapäinen pysähdys.

```
25 PRINT A¤(K),B¤(K),C(K)
35 STOP
```

Aja ohjelma jälleen. Tulokseksi saadaan



Lisätty PRINT-lause helpottaa virheen etsimistä

Kaikkien vaihtoehtojen tarkistaminen

Ennen ohjelman lopputarkistusta poista tilapäiset rivit 25 PRINT A¤(K), B¤(K), C(K) ja 35 STOP.

Palaa takaisin vuokaavion tarkasteluun ja ota selvää, mitä vielä on tarkistettava tässä ohjelmassa. Auton merkkiä koskeva vaihtoehto 2 ja väriä koskeva vaihtoehto 3 eivät ole vielä tarkistetut.

Kotkan
Kaupunginkirjasto

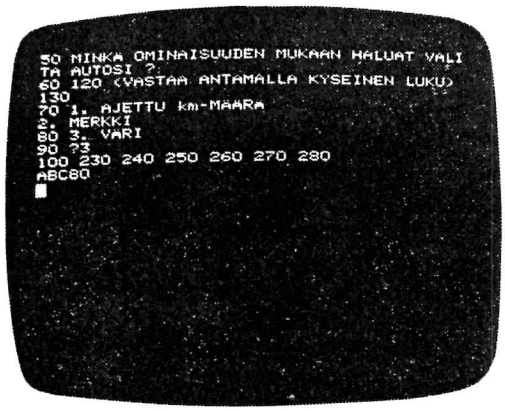
Minkä tulostuksen koeajo antaa? Vaihtoehto 2 toimii mainiosti. Näytöksi saadaan

Vaihtoehdon 2 ajo

Kun kokeilemme vaihtoehtoa 3, tapahtuu jotain odottamatonta. Tietokone ei anna aiottua tulostusta!

Anna komento TRACE ja aja uudestaan!

Tulokseksi saadaan nyt



Vaihtoehdon 3 ajo TRACE-tilassa

Rivin 100 käsky määrää siis hypyn riville 230. Koska haluttu teksti on rivillä 220, täytyy rivi 100 muuttaa muotoon

```
100 ON A GOSUB 140,210,220
```

Kun annat nyt käskyn NOTRACE ja suoritat koeajon, ohjelma toimii myös vaihtoehdon 3 osalta.

Omat virheilmoitukset

Kun taitava ohjelmoija kehittää ohjelmaa, ei ole mitään haittaa siitä, että tietokone antaa virheilmoituksen koodimuodossa. Tilanne on toinen, jos ohjelmaa ajaa asiaan perehtymätön käyttäjä, joka ei osaa tulkita koodeja.

Koska esiintyviä virheitä voidaan usein ennakoida, on mahdollista varustaa ohjelma teksteillä, jotka selittävät syntyneiden virheiden syytä. Käyttäjälle täytyy antaa myös ohjeet virheiden korjaamiseksi.

Olettakaamme, että ajat ohjelmamme uudestaan ja vastaat yksinkertaisella virheellä ajettu km-määrä luvun 1 sijasta tietokoneen ensimmäiseen kysymykseen.

Kiireinen autonostaja hämmentyy melkoisesti, jos hänen toiveeseensa vastataan

```
ERR 12 LINE 90
```

Pienen ohjelmanlisäyksen avulla voidaan asiakkaan tämänkaltainen kohtelu välttää. Lisää aluksi seuraavat rivit ja aja ohjelma.

```
55 ON ERROR GOTO 500
500 STOP
```

Ajo antaa tulostuksen

```
STOP LINE 500
```

On siis mahdollista käskeä tietokonetta hyppäämään riville 500 virheen esiintyessä. Käsky ON ERROR GOTO ... määrää hypyn annetulle riville vain niissä virhetapauksissa, jotka on liitteessä 1 merkitty tähdellä (*).

Korvaa rivi 500 nyt riveillä

```
500 PRINT "YRITÄ UUDELLEEN"
510 GOTO 55
```

Varmistu siitä, että ohjelma toimii nyt, vaikka antaisit virheellisiä vastauksia.

Tässä käytetty yksinkertainen menetelmä merkitsee sitä, että tietokone kirjoittaa YRITÄ UUDESTAAN riippumatta tehdyn virheen laadusta. On myös mahdollista muunnella virheilmoitusta tehdyn virheen laadun mukaan. Yksinkertaisinta on ilmoittaa tehdyn virheen koodinumero.

Lisää rivi

```
505 PRINT "VIRHE NR";ERRCODE
```

Virheen sattuessa tietokone kirjoittaa esim.

```
VIRHE NR 12
```

ERRCODE-käskyä voidaan nyt käyttää siten, että tietokone valitsee sopivan virheilmoituksen.

Yhteenveto

Käsky	Selitys	Esimerkki
TRACE	Suoritettujen ohjelmarivien numerot kirjoitetaan näyttöön ajon aikana.	130 TRACE
NOTRACE	Poistaa käskyn TRACE.	160 NOTRACE
STOP	Lopettaa ohjelmanajon, antaa äänimerkin ja kirjoittaa näyttöön STOP-lauseen rivinumeron. Muuttujia ei nollata.	50 STOP

ON ERROR GOTO...	Aiheuttaa hypyn annetulle riville, jos ajon aikana esiintyy virhe. Vain tähdellä merkityt virheet (liite 2) aiheuttavat hypyn. Kaikki muut virheet antavat normaalin virheilmoituksen.	10 ON ERROR GOTO 500

Komento	Selitys	Esimerkki
TRACE	Suoritettujen ohjelmarivien numerot kirjoitetaan näyttöön ajon aikana.	TRACE
NOTRACE	Poistaa käskyn TRACE.	NOTRACE

Funktio	Selitys	Esimerkki
ERRCODE	Tulostaa viimeisimmän virhekoodin.	500 A=ERRCODE 520 PRINT ERRCODE

Harjoituksia

13.1 Tämän luvun ohjelmaesimerkissä on oma aliohjelmansa vaihtoehdolle 2. Siinä on toistaiseksi vain yksi PRINT-lause. Täydennä tämä aliohjelma siten, että se vastaa vaihtoehtovalintaa 2. Tietokone voi esim. kysyä "MINKÄ AUTOMERKIN HALUAT?" Kun se saa vastauksen tähän kysymykseen, pitää kaikki mainitun merkkiset autot tulostaa kuvaruutuun. Kaikista autoista on merkki-, väri- ja ajomäärätiedot.

13.2 Täydennä ohjelmaesimerkkiä edelleen siten, että kehittelet oman aliohjelman vaihtoehdolle 3, joka koskee autojen väriä.

13.3 Miten edellistä ohjelmaesimerkkiä pitää muuttaa, jotta autokauppias voisi pitää 53 auton tiedot tietokoneen muistissa?

KAPPALE 14

Tietokone käsittelee matriiseja

ABC80

MATRIISILASKENTAA

Taulukot

Olet jo luvussa 8 oppinut, kuinka voidaan sijoittaa ns. kenttä, jossa yksi tai kaksi indeksiä. Yksi-indeksistä kenttää sanotaan vektoriksi, kun taas kaksi-indeksistä kenttää sanotaan taulukoksi eli matriisiksi.

Vuosimalli / Valmistaja	1976	1977	1978	1979	1980
Volvo	4	8	6	6	2
Saab	1	7	3	3	1
Ford	8	7	5	6	2
Volkswagen	1	2	2	2	0

Käyttäkäämme ylläolevasta taulukosta nimeä A. Varjostettu luku on kolmannessa rivissä ja neljännessä sarakkeessa. Siten se saa merkinnän A(3,4), missä A(3,4) = 6. Jokaista taulukon lukua kutsutaan alkioksi eli elementiksi.

Taulukon eli matriisin alkiot voivat olla liukulukuja, kokonaislukuja tai merkkijonoja. Vektoriin tai matriisiin mahtuvien alkioiden lukumäärä määritellään käskyllä DIM.

Jos vektorin tai matriisin kokoa ei ole määritelty, on tilaa varattuna automaattisesti kuten käskyillä DIM A(10) tai DIM A(10, 10).

Seuraava ohjelma lukee INPUT-lauseella yllä olevan matriisin ja tulostaa sen riveittäin.

```
10 FOR I=1 TO 5
20 PRINT "SARAKE";I
30 FOR J=1 TO 4
40 INPUT A(J,I)
```

```
50 NEXT J
60 PRINT
70 NEXT I
80 PRINT "TAULUKKO A"
90 FOR J=1 TO 4
100 FOR I=1 TO 5
110 PRINT A(J,I);
120 NEXT I
130 PRINT
140 NEXT J
```

Kirjoita ja aja ohjelma. Syötä taulukko sarakkeittain ja tarkista, että tulostus on virheetön.

Yllä olevaa matriisien lukemistapaa sarakkeittain ja tulostusta riveittäin käytämme hyväksi tämän luvun matriisilaskuissa.

Matriisien yhteenlasku

Kahden matriisin summa on matriisi, jonka alkiot saadaan siten, että lasketaan keskenään yhteen ne yhteenlaskettavien matriisien alkiot, joilla on sama rivi- ja sarakenumero.

$$\begin{bmatrix} 1 & 7 & 8 \\ 3 & 3 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 8 & 9 \\ 5 & 4 & 4 \end{bmatrix}$$

$$A \quad + \quad B \quad = \quad C$$

Seuraava ohjelma laskee matriisien A ja B summan ja kirjoittaa tuloksen. Rivien määrä R ja sarakkeiden määrä S määritellään INPUT-lauseella. Yhteenlaskettavissa tulee olla sama määrä kumpiakin. Matriisit A ja B määritellään sarakkeittain INPUT-lauseilla. Huomaa, että ohjelmassa ei ole DIM-lausetta. Tarvittaessa on sinun lisättävä se itse.

```
10 PRINT "MATRIISIEN YHTEENLASKU"
20 PRINT
30 PRINT "MATRIISIEN KOKO ( R,S )";
40 INPUT R,S
50 PRINT
60 PRINT "MATRIISI A:"
70 FOR I=1 TO S
80 PRINT "SARAKE";I
90 FOR J=1 TO R
100 INPUT A(J,I)
110 NEXT J
120 PRINT
130 NEXT I
140 PRINT "MATRIISI B:"
150 FOR I=1 TO S
160 PRINT "SARAKE";I
170 FOR J=1 TO R
180 INPUT B(J,I)
190 NEXT J
200 PRINT
210 NEXT I
220 PRINT
230 PRINT
240 PRINT "MATRIISIEN SUMMA ON :"
250 FOR J=1 TO R
260 FOR I=1 TO S
270 N=A(J,I)+B(J,I)
280 PRINT N;
290 NEXT I
300 PRINT
310 NEXT J
```

Matriisien vähennyslasku

Kahden matriisin erotus saadaan laskemalla toisiaan vastaavissa paikoissa olevien alkioiden erotukset.

$$\begin{bmatrix} 1 & 7 & 8 \\ 3 & 3 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 6 & 7 \\ 1 & 2 & 0 \end{bmatrix}$$

$$A \quad - \quad B \quad = \quad C$$

Matriisien vähennyslaskuohjelma saadaan helposti muuttamalla edellisen ohjelman rivit 10, 240 ja 270 muotoon

```
10 PRINT "MATRIISIEN EROTUS"
240 PRINT "MATRIISIEN EROTUS ON :"
270 N=A(J,I)-B(J,I)
```

Matriisien kertolasku

Kahden matriisin tulo määritellään seuraavan esimerkin mukaisesti.

$$\begin{bmatrix} x & x & x \\ a & b & c \\ x & x & x \\ x & x & x \end{bmatrix} * \begin{bmatrix} x & d \\ x & e \\ x & f \end{bmatrix} = \begin{bmatrix} x & x \\ x & g \\ x & x \\ x & x \end{bmatrix}$$

$$A \quad * \quad B \quad = \quad C$$

4 riviä	3 riviä	4 riviä
3 saraketta	2 saraketta	2 saraketta

Alkio g = C(2,2) saadaan matriisien A ja B varjostetuista alkioista kaavalla

$$g = a \cdot d + b \cdot e + c \cdot f$$

Muut alkiot saadaan vastaavalla tavalla.

Huomaa, että matriisin A sarakkeiden määrän tulee olla sama kuin matriisin B rivien määrän.

Seuraava ohjelma laskee kahden matriisin A ja B tulon ja kirjoittaa vastauksen. Matriisin rivien ja sarakkeiden määrät määritellään kahdella INPUT-lauseella. Matriisit A ja B määritellään sarakkeittain INPUT-lauseilla.

```
10 PRINT "MATRIISIEN TULO"
20 PRINT
30 PRINT "MATRIISIN A ULOTTUVUUS
(R,S):";
40 INPUT R1,S1
50 PRINT "MATRIISIN B ULOTTUVUUS
(R,S):";
60 INPUT R2,S2
70 IF S1=R2 THEN 100
80 PRINT "MATRIISEJA A JA B EI VOI
KERTOA ."
81 PRINT "VÄÄRÄ ULOTTUVUUS !"
90 GOTO 20
100 PRINT
110 PRINT "MATRIISI A :"
120 FOR I=1 TO S1
130 PRINT "SARAKE";I
140 FOR J=1 TO R1
150 INPUT A(J,I)
160 NEXT J
170 PRINT
180 NEXT I
190 PRINT
200 PRINT "MATRIISI B :"
210 FOR I=1 TO S2
220 PRINT "SARAKE";I
230 FOR J=1 TO R2
240 INPUT B(J,I)
250 NEXT J
260 PRINT
270 NEXT I
280 PRINT
290 PRINT
300 PRINT "MATRIISIEN TULO ON :"
310 FOR J=1 TO R1
320 FOR I=1 TO S2
330 S=0
340 FOR N=1 TO R2
350 S=S+A(J,N)*B(N,I)
360 NEXT N
```

```
370 PRINT S;
380 NEXT I
390 PRINT
400 NEXT J
```

Matriisin kääntäminen

Matriisin A käänteismatriisia merkitään A^{-1}. Määritelmänä on

$$A^{-1} * A = E$$

jossa E on ns. yksikkömatriisi. Sen alkiot ovat nollia lukuunottamatta yhtä lävistäjää, jonka kaikki alkiot ovat ykkösiä.

Esimerkki:

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Jotta matriisi voitaisiin kääntää, täytyy sen rivien ja sarakkeiden lukumäärien olla samoja.

Seuraava ohjelma laskee A:n käänteismatriisin ja antaa tuloksen kolmen desimaalin tarkkuuteen pyöristettynä. Rivien ja sarakkeiden yhteinen määrä annetaan INPUT-lauseella. Matriisi A määritetään sarakkeittain INPUT-lauseella.

```
10 PRINT "KÄÄNTEISMATRIISI"
20 PRINT
30 PRINT "MATRIISIN ULOTTUVUUS R :";
40 INPUT R
50 PRINT
60 PRINT "MATRIISI A :"
70 FOR I=1 TO R
80 PRINT "SARAKE";I
90 FOR J=1 TO R
100 INPUT A(J,I)
110 NEXT J
120 PRINT
```

```
130 B(I,I)=1
140 NEXT I
150 FOR J=1 TO R
160 FOR I=J TO R
170 IF A(I,J)<>0 THEN 210
180 NEXT I
190 PRINT "MATRIISI EI OLE KÄÄNTEINEN"
200 GOTO 20
210 FOR K=1 TO R
220 S=A(J,K)
230 A(J,K)=A(I,K)
240 A(I,K)=S
250 S=B(J,K)
260 B(J,K)=B(I,K)
270 B(I,K)=S
280 NEXT K
290 T=1/A(J,J)
300 FOR K=1 TO R
310 A(J,K)=T*A(J,K)
320 B(J,K)=T*B(J,K)
330 NEXT K
340 FOR L=1 TO R
350 IF L=J THEN 410
360 T=-A(L,J)
370 FOR K=1 TO R
380 A(L,K)=A(L,K)+T*A(J,K)
390 B(L,K)=B(L,K)+T*B(J,K)
400 NEXT K
410 NEXT L
420 NEXT J
430 PRINT
440 PRINT
450 PRINT "KÄÄNTEISMATRIISI :"
460 FOR I=1 TO R
470 FOR J=1 TO R
480 PRINT INT(B(I,J)*1000+.5)/1000;
490 NEXT J
500 PRINT
510 NEXT I
```

Yhtälöryhmän ratkaiseminen

Tarkastele seuraavaa yhtälöryhmää

$$\begin{cases} a_{11}\,x_1 + a_{12}\,x_2 \;\cdots\cdots\; a_{1n}\,x_n = c_1 \\ a_{21}\,x_1 + a_{22}\,x_2 \;\cdots\cdots\; a_{2n}\,x_n = c_2 \\ \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\quad \vdots \\ a_{n1}\,x_1 + a_{n2}\,x_2 \;\cdots\cdots\; a_{nn}\,x_n = c_n \end{cases}$$

missä x_1 — x_n ovat tuntemattomia. Matriisien avulla voidaan yhtälöryhmä kirjoittaa seuraavalla tavalla:

$$A * X = C$$

missä

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Matriisimuodossa kirjoitettuna tulee yhtälöryhmän ratkaisuksi $X = A^{-1} * C$ eli A käännetään ja kerrotaan sitten C:llä.

Seuraava ohjelma määrittää yhtälöryhmän $A * X = C$ ratkaisun ja kirjoittaa tuloksen. Tuntemattomien lukumäärä määritellään INPUT-lauseella. Matriisi A ja vektori C määritellään INPUT-lauseilla.

```
10 PRINT "YHTÄLÖRYHMÄN RATKAISU"
20 PRINT
30 PRINT "TUNTEMATTOMIEN LUKUMÄÄRÄ :";
40 INPUT R
50 PRINT
60 PRINT "KERROINMATRIISI A :"
70 FOR I=1 TO R
80 PRINT "SARAKE";I
```

```
90 FOR J=1 TO R
100 INPUT A(J,I)
110 NEXT J
120 PRINT
130 B(I,I)=1
140 NEXT I
150 PRINT "VAKIOVEKTORI C :"
160 FOR L=1 TO R
170 INPUT Y(L)
180 NEXT L
190 FOR J=1 TO R
200 FOR I=J TO R
210 IF A(I,J)<>0 THEN 250
220 NEXT I
230 PRINT "YHTÄLÖRYHMÄLLÄ EI OLE
RATKAISUA."

240 GOTO 20
250 FOR K=1 TO R
260 S=A(J,K)
270 A(J,K)=A(I,K)
280 A(I,K)=S
290 S=B(J,K)
300 B(J,K)=B(I,K)
310 B(I,K)=S
320 NEXT K
330 T=1/A(J,J)
340 FOR K=1 TO R
350 A(J,K)=T*A(J,K)
360 B(J,K)=T*B(J,K)
370 NEXT K
380 FOR L=1 TO R
390 IF L=J THEN 450
400 T=-A(L,J)
410 FOR K=1 TO R
420 A(L,K)=A(L,K)+T*A(J,K)
430 B(L,K)=B(L,K)+T*B(J,K)
440 NEXT K
450 NEXT L
460 NEXT J
470 PRINT
480 PRINT
490 PRINT "YHTÄLÖRYHMÄN RATKAISU :"
500 FOR I=1 TO R
```

```
510 S=0
520 FOR K=1 TO R
530 S=S+B(I,K)*Y(K)
540 NEXT K
550 PRINT "X (";I;" ) =";S
560 NEXT I
```

Yhteenveto

Käsky	Selitys	Esimerkki
DIM	Varaa tilaa kenttämuuttujalle. Esimerkkejä:	
	Varaa tilan 26 liukulukumuuttujalle väliltä A(0)—A(25).	`10 DIM A(25)`
	Varaa tilan merkkijonolle A¤, jossa on enintään 200 merkkiä.	`20 DIM A¤=200`
	Varaa tilan 31 jonomuuttujalle A (0)—A (30) siten, että kussakin voi olla 20 merkkiä.	`30 DIM A¤(30)=20`
	Varaa tilan kokonaislukumuuttujamatriisille, jossa on 11 riviä ja 21 saraketta.	`40 DIM A%(10,20)`
	Varaa tilan merkkijonomatriisille, jossa on 5 riviä ja 6 saraketta siten, että jokaisessa alkiossa voi olla enintään 30 merkkiä.	`50 DIM A¤(4,5)=30`

LIITE 1

Tietokoneen virheilmoitukset

ABC80

Virheilmoitukset

ERROR 0 RAJOJEN YLITYS. Matriisia ei voi kasvattaa yli määritellyn koon.

ERROR 1 VIRHEELLINEN INDEKSIEN MÄÄRÄ. Indeksien lukumäärä erilainen kuin DIM-lauseessa määritelty.

ERROR 2 VIRHEELLINEN KOMENTO. Käskyä voidaan käyttää vain ohjelmassa eikä yksinään komentona.

ERROR 3 MUISTI TÄYSI. Käytettävissä oleva muistitila on kokonaan varattu.

ERROR +4 LIIAN SUURI LIUKULUKU.

ERROR 5 LIIAN SUURI INDEKSI. Yritys käyttää suurempaa indeksiä kuin DIM-lauseessa määritelty.

ERROR 6 PUUTTUVA RIVINUMERO. Viittaus rivinumeroon, jota ohjelmasta ei löydy.

ERROR +7 LIIAN SUURI KOKONAISLUKU. Yritys käyttää kokonaislukua, jonka arvo on tietokoneen käsittelykyvyn ulkopuolella.

ERROR 8 EI KÄYTÖSSÄ TÄSSÄ JÄRJESTELMÄSSÄ. Virheilmoitus esiintyy vain ohjelmien kehitystyön aikana.

ERROR 9 INDEKSIN ARVO JONON ULKOPUOLELLA. Indeksi on liian suuri tai negatiivinen.

ERROR 10 TEKSTI EI MAHDU MERKKIJONOON. Merkkijonon dimensio on määritelty liian pieneksi.

ERROR 11 MUODOLLINEN VIRHE. Annettua lausetta ei tunnisteta käskyksi eikä komennoksi.

ERROR +12 VIRHEELLINEN LUKU. Luku sisältää merkkejä, joita luvuissa ei sallita.

ERROR 13 VIRHEELLINEN ARGUMENTTIEN LUKUMÄÄRÄ TAI LAATU.

ERROR 14 VIRHEELLINEN MERKKI LAUSEEN PERÄSSÄ. Muodollinen virhe. Tietokone odotti seuraavaksi merkiksi RETURN tai kaksoispiste.

ERROR 15 "=" MERKKI PUUTTUU TAI ON VÄÄRÄSSÄ PAIKASSA. FOR ja LET lauseissa esiintyvä muodollinen virhe.

ERROR 16 PUUTTUVA RIVINUMERO. Ohjelmassa on jokin ohjelmarivi ilman rivinumeroa.

ERROR 17 VIRHEELLINEN LUKUJEN JA MERKKIJONOJEN YHDISTELMÄ. Merkkijonoja ja lukuja ei voida käsitellä samassa operaatiossa.

ERROR +18 ")" MERKKI PUUTTUU TAI ON VÄÄRÄLLÄ PAIKALLA. Muodollinen virhe.

ERROR +19 TIEDOSTON AVAAMINEN EI OLE SALLITTU. Liian monta tiedostoa avattuna yhtäaikaa.

ERROR 20 LIIAN PITKÄ RIVI (>120 merkkiä). Suurin sallittu pituus ohjelma- tai datariville on 120 merkkiä.

ERROR +21 TIEDOSTOA EI LÖYDY. Tietokone ei löydä määriteltyä tiedostoa. Tiedosto puuttuu tai tiedoston nimi on väärä.

ERROR 22 KIELLETTY LAUSE.

ERROR +23 "TO" SANA PUUTTUU. Virhe esiintyy vain FOR lauseissa.

ERROR 24 "NEXT" SANA PUUTTUU. Ohjelmassa on FOR lause, jota vastaava NEX puuttuu.

ERROR 25 VIRHEELLINEN LAUSE "ON" SANAN JÄLKEEN. Muodollinen virhe.

ERROR 26 VIRHEELLINEN ON-LAUSE.

ERROR 27 "NEXT" ILMAN "FOR" SANAA. Ohjelmassa on NEXT-lause, jota vastaavaa FOR sanaa ei löydy.

ERROR 28 VIRHEELLINEN MUUTTUJA "NEXT" SANAN JÄLKEEN. Muodollinen virhe.

ERROR 29 "RETURN" ILMAN "GOSUB" LAUSETTA. Ohjelmassa on RETURN lause ilman edeltävää GOSUB aliohjelmakutsua.

ERROR +30 TIEDOT LOPPU. Tietolista on käytetty loppuun ja ohjelmassa on READ lause, jolla luetaan lisää tietoa.

ERROR 31 VIRHEELLINEN TIETO KOMENNOSSA. Komentoon on liitetty virheellinen argumentti esim. UNSAVE komento ilman tiedostonimeä.

ERROR 32 TIEDOSTOA EI AVATTU. INPUT tai PRINT lause kohdistettu avaamattomaan tiedostoon.

ERROR 33 "AS FILE" PUUTTUU. Virhe esiintyy OPEN ja PREPARE lauseissa.

ERROR +34 TIEDOSTO LOPPU. Yritys lukea tiedostosta loppumerkin jälkeen.

ERROR +35 TARKISTUSSUMMAVIRHE LUETTAESSA. Kalvolevy tai kasetti vioittunut.

ERROR +36 x TARKISTUSSUMMAVIRHE KIRJOITTAESSA. Kalvolevy vahingoittunut.

ERROR +37 VIRHEELLINEN TALLENNUSFORMAATTI. Tallennusformaatti ei ole ABC 80 -mikrotietokoneessa käytetty. Virhe voi myös merkitä levyssä tai kasetissa olevaa vikaa.

ERROR +38 TIETUENUMERO TIEDOSTON ULKOPUOLELLA. Yritys lukea enemmän, kuin tiedostossa on tietoa.

ERROR +39 x KIRJOITUSSUOJATTU TIEDOSTO. Tiedosto on suojattu kirjoittamiselta.

ERROR +40 x TYHJENNYSSUOJATTU TIEDOSTO. Tiedosto on suojattu tyhjentämiseltä.

ERROR +41 x LEVY TÄYSI. Kalvolevyllä ei ole tilaa uusille tiedostoille.

ERROR +42 x LEVY-YKSIKKÖ EI VALMIS. Levy-yksikössä ei ole levyä tai levy-yksikön kansi on auki.

ERROR +43 x LEVY ON KIRJOITUSSUOJATTU.

ERROR 44 x LOOGISTA TIEDOSTOA EI OLE AVATTU KÄYTTÖÄ VARTEN.

ERROR 45 x VIRHEELLINEN LOOGINEN TIEDOSTONUMERO.

ERROR 46 VIRHEELLINEN YKSIKKÖNUMERO.

ERROR 47 x VIRHEELLINEN YKSIKKÖNUMERO.

ERROR 48 x VIRHE LEVYKIRJASTOSSA.

ERROR 49 x VIRHEELLINEN FYYSINEN TIEDOSTONUMERO.

ERROR 50 NEGATIIVINEN JUURRETTAVA. Looginen virhe ohjelmassa.

ERROR 51 YKSIKKÖ VARATTU.

ERROR 52 VIRHEELLINEN TOIMINTO YKSIKÖLLE. Esimerkiksi INPUT lause kohdistettuna kirjoittimeen PR:

ERROR 53 VIRHEELLINEN RIVI.

ERROR 57 MÄÄRITTELEMÄTÖN FUNKTIO. Viittaus funktioon, jota ei ole määritelty DEF lauseessa.

ERROR 58 VIRHEELLINEN MERKKI LUETTU.

ERROR 59 VÄÄRÄ OHJAUSFORMAATTI. Ohjelma on tallennettu BASIC-muunnokseen, joka ei ole yhteensopiva käytetyn kielen kanssa.

ERROR 60 BITTIOSOITE 16 BITTIÄ.

ERROR 61 PUUTTUVA PILKKU.

ERROR 62 GRAFIIKKAPISTEEN OSOITE KUVARUUDUN ULKOPUOLELLA.

ERROR 63 PUUTTUVA "AS" SANA. Virhe NAME AS lauseessa.

ERROR 64 VIRHEELLINEN "RENAME" LAUSE. Uusi tiedostonimi ei ole pätevä.

ERROR 65 ASCII-ARITMETIIKKAVIRHE.

ERROR 66 EI-NUMEERINEN MERKKIJONO.

HUOM!: x tarkoittaa levyvirhettä, o tarkoittaa valinnaista yksikköä ja + tarkoittaa virhettä, jota voidaan käsitellä ON ERROR GOTO lauseella.

LIITE 2

Tietokoneen vastaukset harjoitustehtäviin

ABC80

2.1
```
MITÄ
TÄMÄ
TULOSTAA ?
```

2.2
```
ENTÄ            SITTEN          TÄMÄ ?
```

2.3
```
10 PRINT "MITEN ";
20 PRINT "TÄMÄ ";
30 PRINT "TULOSTETAAN ?"
```

2.4
```
ANNA PAKKALA
PÄÄKATU 10
02100 ESPOO 10
```

2.5
```
                    LIISA PAKKALA
                    PÄÄKATU 10
                    02100 ESPOO 10
```

2.6
```
10 GOTO 40
20 PRINT "K";
30 GOTO 60
40 PRINT "O";
50 GOTO 20
60 PRINT "!"
```

2.7 Sarakkeeseen 3

3.1
```
5 PRINT
6 PRINT "KENEN PUHELINNUMERON HALUAT ?"
7 INPUT N¤
10 READ A¤,B¤
20 DATA "PEKKA","123456"
21 DATA "OLLI","234567"
22 DATA "ERKKI","122113"
23 DATA "KAUKO","433415"
24 DATA "LEENA","545325"
29 DATA "KAARINA","345678"
30 IF N¤=A¤ THEN GOTO 50
35 IF A¤="KAARINA" THEN GOTO 90
40 GOTO 10
50 PRINT "NUMERO ON",B¤
60 GOTO 100
90 PRINT "EI LÖYDY LUETTELOSTAMME !"
100 PRINT "SELVÄ"
```

3.2
```
5 PRINT
6 PRINT "ANNA HALUTTU PUHELINNUMERO :"
7 INPUT N¤
10 READ A¤,B¤
20 DATA "PEKKA","123456"
21 DATA "OLLI","234567"
22 DATA "ERKKI","122113"
23 DATA "KAUKO","433415"
24 DATA "LEENA","545325"
29 DATA "KAARINA","345678"
30 IF N¤=B¤ THEN GOTO 50
35 IF A¤="KAARINA" THEN GOTO 90
40 GOTO 10
50 PRINT "NIMI ON",A¤
60 GOTO 100
90 PRINT "EI LÖYDY LUETTELOSTAMME !"
100 PRINT "SELVÄ"
```

3.3 a
```
160 DATA "VOLVO","PUNAINEN","30000"
170 DATA "TOYOTA","SININEN","60000"
```

3.3 b POISTA RIVI 130

3.4
```
10 PRINT
20 PRINT "HALUTTU AUTON VÄRI ?"
30 INPUT A$
40 PRINT
50 PRINT "MERKKI","VÄRI","AJETTU (km)"
60 PRINT
70 READ B$,C$,D$
80 IF A$=C$ THEN GOTO 200
90 IF B$="LOPPU" THEN GOTO 220
100 GOTO 70
110 DATA "FIAT","SININEN","40000"
120 DATA "SAAB","VIHREÄ","15000"
130 DATA "VOLVO","VIHREÄ","10000"
140 DATA "SAAB","KELTAINEN","20000"
150 DATA "VOLVO","PUNAINEN","120000"
160 DATA "VOLVO","PUNAINEN","35000"
170 DATA "TOYOTA","SININEN","60000"
190 DATA "LOPPU","LOPPU","LOPPU"
200 PRINT B$,C$,D$
210 GOTO 70
220 PRINT
```

4.1 a
```
10 PRINT (2+5)*3
```

4.1 b
```
20 PRINT (2+5)/7
```

4.1 c
```
30 LET A=2
40 LET B=3
50 PRINT ((A+B)*A*B)/A
```

4.2
```
10 READ A,B,C,D,E
20 LET S=A+B+C+D+E
30 DATA 2,4,7,11,16
50 PRINT "SUMMA ON",S
```

4.3 4 2

4.4 Mikäli perheenjäsentesi lukumäärä on myös 4 henkeä, ei sinun tarvitse muuttaa ED-komennolla kuin rivi 30. Mikäli myös lukumäärä on eri kuin tehtävässä, on sinun muutettava myös rivit 10 ja 20 vastaaviksi.

4.5
```
10 PRINT "ANNA AJETTU MATKA (km) ";
20 INPUT S
30 PRINT "PALJONKO KÄYTETTY BENSAA? (l)";
40 INPUT L
50 PRINT "KULUTUS=";L/S*100;" LITRAA/100
KILOMETRIÄ"
55 PRINT
60 GOTO 10
```

4.6
```
10 PRINT "ILMOITA LUKU ";
20 INPUT X
30 PRINT
40 PRINT "LUVUN";X;" KERTOTAULU ON SEURAAVA:"
50 LET A=A+1
60 PRINT X;" *";A;"=";X*A
70 IF A=12 THEN GOTO 90
80 GOTO 50
90 PRINT
```

4.7
```
10 PRINT "KUINKA MONTA PERHEENJÄSENTÄ",
20 INPUT N
30 LET A=A+1
40 PRINT "KUINKA VANHA ON JÄSEN ";A,
50 INPUT X
60 LET S=S+X
70 IF A=N THEN GOTO 100
80 GOTO 30
100 LET K=S/N
110 PRINT "PERHEEN KESKI-IKÄ ON";K;" VUOTTA!"
```

5.1
```
10 FOR I=1 TO 20
20 PRINT "ÄITI SOUTAA JA ISÄ VETÄÄ UISTINTA"
30 NEXT I
```

5.2
```
10 PRINT CHR$(12)
20 RANDOMIZE
30 PRINT
40 PRINT "VOITTOLISTA"
50 PRINT
60 FOR I=1 TO 10
70 LET V=INT(100*RND)
80 PRINT "VOITTO ";I,"NUMERO ";V
90 NEXT I
```

5.3
```
10 PRINT CHR$(12)
20 RANDOMIZE
30 PRINT
40 PRINT "VOITTOLISTA"
50 PRINT
60 FOR I=1 TO 15
70 LET V=100+INT(200*RND)
80 PRINT "VOITTO ";I,"NUMERO ";V
90 NEXT I
```

5.4
```
10 RANDOMIZE
20 PRINT
30 LET N=INT(1+6*RND)
40 PRINT "NOPPA PYSÄHTYI , TULOS ON ";N
```

5.5
```
10 RANDOMIZE
20 LET A=INT(100*RND)
30 LET B=INT(100*RND)
40 PRINT
50 PRINT "LUVUT OVAT ";A;" JA ";B
60 PRINT "MIKÄ ONKAAN NIIDEN SUMMA ";
70 INPUT S
80 IF S=A+B THEN 130
90 PRINT
100 PRINT "??? VÄÄRIN ???"
110 PRINT
120 GOTO 50
130 PRINT
140 PRINT "%%% OIKEIN %%%"
```

5.6
```
10 PRINT
20 FOR I=1 TO 10
30 PRINT "PROFESSORI No ";I
40 PRINT "OIKEITA VASTAUKSIA ";
50 INPUT P(I)
60 PRINT
70 LET S=S+P(I)
80 NEXT I
90 LET K=S/10
100 PRINT
110 FOR I=1 TO 10
120 PRINT "PROF. No ";I,"OIKEIN ";P(I)
130 NEXT I
140 PRINT
150 PRINT "VASTAUKSIEN KESKIARVO : ";K;" OIKEIN"
```

5.7
```
10 PRINT
20 PRINT "ARVAA LASKEMANI SATUNNAISLUKU
( 0 - 99 )"
30 PRINT
40 RANDOMIZE
50 LET S=INT(RND*100)
60 PRINT "ARVAUKSESI ";
70 INPUT A
80 LET F=F+1
90 IF A=S THEN GOTO 170
100 IF A>S THEN GOTO 140
110 PRINT "LIIAN PIENI"
120 PRINT
130 GOTO 60
140 PRINT "LIIAN SUURI"
150 PRINT
160 GOTO 60
170 PRINT "IHAN  O I K E I N  ARVATTU !"
180 PRINT "YRIYKSIÄ OLI YHTEENSÄ ";F
```

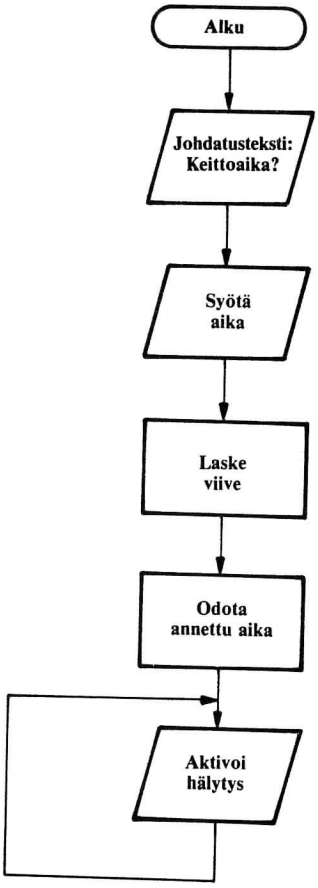
5.8 a Todennäköisyydet :

ykkönen	0,6
risti	0,2
kakkonen	0,2

5.8 b
```
10 PRINT CHR$(12)
20 RANDOMIZE
30 PRINT "VEIKKAUSRIVI"
40 PRINT
50 FOR X=1 TO 13
60 PRINT "OTTELU No";X,
70 T=RND
80 IF T<.5 THEN 130
90 IF T<.75 THEN 150
100 PRINT "      2"
110 NEXT X
120 GOTO 170
130 PRINT " 1"
140 GOTO 110
150 PRINT "    X"
160 GOTO 110
170 PRINT
180 PRINT "ONNEA YRITYKSELLESI"
```

5.9
```
10 PRINT CHR$(12)
20 FOR I=1 TO 7
30 PRINT "ETSIVÄ No ";
40 INPUT E$(I)
50 PRINT "ÄO ";
60 INPUT Ä(I)
70 NEXT I
80 FOR I=1 TO 6
90 FOR J=I+1 TO 7
100 IF Ä(I)>Ä(J) THEN 170
110 X=Ä(I)
120 Ä(I)=Ä(J)
130 Ä(J)=X
140 Y$=E$(I)
150 E$(I)=E$(J)
160 E$(J)=Y$
170 NEXT J
180 NEXT I
190 PRINT CHR$(12)
200 PRINT "AGENTIKSI PÄÄSI ETSIVÄ    ";E$(1)
210 PRINT
220 PRINT "VARALLA OVAT SEURAAVASSA
JÄRJESTYKSESSÄ"
230 PRINT
240 FOR I=2 TO 7
250 PRINT "ETSIVÄ No:";E$(I)
260 NEXT I
```

7.1
```
10 PRINT "MIKÄ ON HALUAMASI KEITTOAIKA
MINUUTEISSA"
20 INPUT M
30 T=M*1000*60
40 FOR L=1 TO T
50 NEXT L
60 PRINT CHR$(7)
70 GOTO 60
```



7.2 Noin 35 vuotta.

7.4
```
10 LET N=1
20 OUT 6,N
30 FOR V=1 TO 1000 : NEXT V
40 N=N+2
50 IF N<256 THEN 20
60 END
```

7.5
```
10 GET A¤
20 IF A¤="A" THEN GOSUB 90
30 IF A¤="B" THEN GOSUB 100
40 IF A¤="C" THEN GOSUB 110
50 IF A¤="D" THEN GOSUB 120
60 IF A¤="E" THEN GOSUB 130
70 IF A¤="F" THEN GOSUB 140
80 GOTO 10
90 OUT 6,5 : RETURN
100 OUT 6,3 : RETURN
110 OUT 6,251 : RETURN
120 OUT 6,19 : RETURN
130 OUT 6,9 : RETURN
140 OUT 6,39 : RETURN
```

8.1
```
10 PRINT
20 PRINT "MIKÄ SINUN NIMESI ON"
30 INPUT A¤
40 PRINT
50 PRINT "HEI ";A¤
60 PRINT A¤;" ON KIVA NIMI"
```

8.2
```
10 A¤="OHJELMOINNIN ABC KIRJA"
20 PRINT
30 FOR I=1 TO 22
40 IF MID¤(A¤,I,1)=" " THEN PRINT " -";
50 PRINT MID¤(A¤,I,1);
60 NEXT I
70 PRINT
```

8.3
```
10 PRINT "KIRJOITA HENKILÖTUNNUKSESI"
20 INPUT A¤
30 V¤=MID¤(A¤,5,2)
40 PRINT
50 PRINT "OLET SYNTYNYT VUONNA 19";V¤
```

8.4
```
10 RANDOMIZE
20 FOR X=1 TO 60
30 FOR I=1 TO 2
40 A(I)=INT(65+29*RND)
50 NEXT I
60 PRINT CHR¤(86,A(1),A(2)),
70 NEXT X
```

8.5 Nimet lajitellaan oikeaan järjestykseen vain ensimmäisenä olevien Å,Ä ja Ö kirjainten mukaan .

8.6
```
10 PRINT "KIRJOITA HALUTTU TEKSTI"
20 INPUT A$
30 PRINT CHR$(12)
40 A=LEN(A$)
50 X$=STRING$(A+2,42)
60 PRINT X$
70 PRINT "*";A$;"*"
80 PRINT X$
```

8.7
```
10 PRINT "SYÖTÄ OSANOTTAJALUETTELO
ARVONIMINEEN "
20 FOR X=1 TO 4
30 INPUTLINE A$(X)
40 PRINT
50 NEXT X
60 PRINT CHR$(12)
70 PRINT "Uudet Y L I V Ä Ä P E L I T:"
80 PRINT
90 PRINT
100 FOR X=1 TO 4
110 PRINT "YLI"+A$(X)
120 NEXT X
```

8.8
```
10 DIM A$=120
20 INPUTLINE A$
30 A=LEN(A$)-2 : S=0
40 FOR X=1 TO A
50 IF MID$(A$,X,1)<>"S" THEN 70
60 S=S+1
70 NEXT X
80 PRINT CHR$(12)
90 F=(S/A)*100
100 PRINT "S:n esiintymisfrekvenssi = ";F;" %"
110 IF F<5 THEN 140
120 OUT 6,7
130 PRINT "*** Kirjoita uudelleen ***"
140 GOTO 10
```

8.9
```
10 DIM A$=120
20 INPUTLINE A$
30 B$="KUSTAA VANHA"
40 X=INSTR(1,A$,B$)
50 IF X=0 THEN 185
60 PRINT
70 PRINT
80 PRINT "VAIHDETAAN"
90 PRINT
100 FOR I=1 TO X-1
110 PRINT MID$(A$,I,1);
120 NEXT I
130 PRINT "KALLE KUSTAA ";
140 FOR I=X+9 TO LEN(A$)
150 PRINT MID$(A$,I,1);
160 NEXT I
170 GOTO 200
180 PRINT
185 PRINT
190 PRINT "HYVÄKSYTÄÄN"
200 GOTO 20
```

9.1
```
10 PRINT CHR$(12)
20 PRINT
30 PRINT "X=";
40 INPUT X
50 LET Y=X↑5%+X↑2%+2
60 PRINT "Y=";Y
70 GOTO 20
```

9.2
```
10 PRINT "X","LOG(X)","EXP(X)"
20 FOR X=10 TO 19
30 PRINT X,LOG(X),EXP(X)
40 NEXT X
```

9.3
```
10 PRINT "X","LOG(X)","EXP(X)"
20 FOR X=10 TO 19 STEP .5
30 PRINT X,LOG(X),EXP(X)
40 NEXT X
```

9.4
```
10 PRINT "LAUSEKKEEN C=SQR(A**2%+B**2%)
LASKENTA"
20 PRINT
30 PRINT "SYÖTÄ A ja B  < A,B > ";
40 INPUT A,B
50 LET C=SQR(AÜ2%+BÜ2%)
60 PRINT "C="C
70 GOTO 20
```

9.5
```
10 PRINT CHR¤(12)
20 PRINT "LAUSEKKEEN (SQR(X))*EXP(-X**2%)
ARVO"
30 PRINT
40 DEFFNA(X)=(SQR(X))*EXP(-XÜ2%)
50 PRINT "X","FNA(X)"
60 READ X
70 IF X=999 THEN 100
80 PRINT X,FNA(X)
90 GOTO 60
100 DATA 1.7,2.25,2.5,999
110 END
```

9.6
```
10 PRINT CHR¤(12)
20 PRINT "NELIÖJUURI KAHDEN DESIM.
TARKKUUDELLA"
30 PRINT
40 PRINT "X=";
50 INPUT X
60 PRINT "SQR(X)=";INT(100*SQR(X)+.5)/100
70 GOTO 30
```

9.7
```
10 PRINT CHR¤(12)
20 FOR X=-10 TO 10 STEP .5
30 LET Y=ATN(X)
40 PRINT CUR(12-7*Y,2*X+20);"*"
50 NEXT X
```

9.8
```
10 PRINT CHR¤(12)
20 FOR X=0 TO 6.4 STEP .4
30 LET Y=SIN(X)
40 PRINT CUR(12-10*ABS(Y),5*X);"*"
50 NEXT X
60 PRINT CUR(20,0);"'KOKOAALLON TASASUUNTAUS'"
```

11.2
```
10 DIM A(20)
20 PRINT "PIENI VIIVE ,KUN TIETO"
30 PRINT "HAETAAN KASETILTA"
40 OPEN "TIEDOSTO" ASFILE 1
50 FOR I=1 TO 20
60 INPUT #1,A(I)
70 NEXT I
80 CLOSE 1
90 FOR I=1 TO 20
100 PRINT A(I)
110 NEXT I
```

12.1
```
10 ; CHR¤(12)
20 ; "MYYNTIHINTA","MYYNTIPALKKIO"
30 ;
40 P¤="0.04"
50 FOR K=1 TO 4
60 READ A¤
70 ; A¤,MUL¤(P¤,A¤,2)
80 NEXT K
90 DATA 524073,175011,377775,212538
```

13.1
```
10 FOR K=1 TO 6
20 READ A¤(K),B¤(K),C(K)
30 NEXT K
40 PRINT CHR¤(12)
50 PRINT "MINKÄ OMINAISUUDEN MUKAAN HALUAT
VALITA AUTOSI ?:"
60 GOSUB 120
70 PRINT "1. AJETTU km-MÄÄRÄ" : PRINT "2.
MERKKI"
80 PRINT "3. VÄRI"
90 INPUT A
100 ON A GOSUB 140,210,260
110 END
120 PRINT "(VASTAA ANTAMALLA KYSINEN LUKU)"
130 PRINT : RETURN
140 PRINT "ILMOITA km-MÄÄRÄ (noin ! )"
150 GOSUB 120
160 INPUT A
170 FOR K=1 TO 6
180 IF ABS(A-C(K))>25000 THEN 200
190 PRINT A¤(K),B¤(K),C(K)
200 NEXT K : RETURN
210 PRINT "MINKÄ AUTOMERKIN HALUAISIT ";
220 INPUT A¤
230 FOR K=1 TO 6
240 IF A¤=A¤(K) THEN PRINT A¤(K),B¤(K),C(K)
250 NEXT K : RETURN
```

```
260 PRINT "KOHTAA 3. KOSKEVA OHJELMAN OSA EI
VIELÄ OLE VALMIS" : RETURN
270 DATA VOLKSWAGEN,SININEN,40000
280 DATA OPEL,VIHREÄ,50000
290 DATA FIAT,KELTAINEN,20000
300 DATA VOLVO,PUNAINEN,70000
310 DATA AUDI,VALKOINEN,80000
320 DATA SAAB,RUSKEA,30000
```

13.2

```
10 FOR K=1 TO 6
20 READ A$(K),B$(K),C(K)
30 NEXT K
40 PRINT CHR$(12)
50 PRINT "MINKÄ OMINAISUUDEN MUKAAN HALUAT
VALITA AUTOSI ?:"
60 GOSUB 120
70 PRINT "1. AJETTU km-MÄÄRÄ" : PRINT "2.
MERKKI"
80 PRINT "3. VÄRI"
90 INPUT A
100 ON A GOSUB 140,210,260
110 END
120 PRINT "(VASTAA ANTAMALLA KYSINEN LUKU)"
130 PRINT : RETURN
140 PRINT "ILMOITA km-MÄÄRÄ (noin ! )"
150 GOSUB 120
160 INPUT A
170 FOR K=1 TO 6
180 IF ABS(A-C(K))>25000 THEN 200
190 PRINT A$(K),B$(K),C(K)
200 NEXT K : RETURN
210 PRINT "MINKÄ AUTOMERKIN HALUAISIT ";
220 INPUT A$
230 FOR K=1 TO 6
240 IF A$=A$(K) THEN PRINT A$(K),B$(K),C(K)
250 NEXT K : RETURN
260 PRINT
270 PRINT "MIKÄ ON VÄRITOIVOMUKSESI";
280 INPUT A$
290 FOR K=1 TO 6
300 IF A$=B$(K) THEN PRINT A$(K),B$(K),C(K)
310 NEXT K : RETURN
320 DATA VOLKSWAGEN,SININEN,40000
330 DATA OPEL,VIHREÄ,50000
340 DATA FIAT,KELTAINEN,20000
350 DATA VOLVO,PUNAINEN,70000
360 DATA AUDI,VALKOINEN,80000
370 DATA SAAB,RUSKEA,30000
```

13.3
```
1 DIM A$(53),B$(53),C(53)
10 FOR K=1 TO 53
20 READ A$(K),B$(K),C(K)
30 NEXT K
40 PRINT CHR$(12)
50 PRINT "MINKÄ OMINAISUUDEN MUKAAN HALUAT
VALITA AUTOSI ?:"
60 GOSUB 120
70 PRINT "1. AJETTU km-MÄÄRÄ" : PRINT "2.
MERKKI"
80 PRINT "3. VÄRI"
90 INPUT A
100 ON A GOSUB 140,210,260
110 END
120 PRINT "(VASTAA ANTAMALLA KYSINEN LUKU)"
130 PRINT : RETURN
140 PRINT "ILMOITA km-MÄÄRÄ (noin ! )"
150 GOSUB 120
160 INPUT A
170 FOR K=1 TO 53
180 IF ABS(A-C(K))>25000 THEN 200
190 PRINT A$(K),B$(K),C(K)
200 NEXT K : RETURN
210 PRINT "MINKÄ AUTOMERKIN HALUAISIT ";
220 INPUT A$
230 FOR K=1 TO 53
240 IF A$=A$(K) THEN PRINT A$(K),B$(K),C(K)
250 NEXT K : RETURN
260 PRINT
270 PRINT "MIKÄ ON VÄRITOIVOMUKSESI";
280 INPUT A$
290 FOR K=1 TO 53
300 IF A$=B$(K) THEN PRINT A$(K),B$(K),C(K)
310 NEXT K : RETURN
320 DATA VOLKSWAGEN,SININEN,40000
330 DATA OPEL,VIHREÄ,50000
340 DATA FIAT,KELTAINEN,20000
350 DATA VOLVO,PUNAINEN,70000
360 DATA AUDI,VALKOINEN,80000
370 DATA SAAB,RUSKEA,30000
380 REM
390 REM ****************************
400 REM ** LISÄÄ TÄHÄN 47 UUTTA **
410 REM ** DATALAUSETTA , JOTTA **
420 REM ** OHJELMA VOISI TOIMIA **
430 REM ****************************
```

14.1 a A+B=
$$\begin{bmatrix} 4 & 3 & 3 \\ -2 & 3 & 3 \\ 5 & 7 & 2 \end{bmatrix}$$

14.1 b A−B=
$$\begin{bmatrix} -2 & 7 & 5 \\ 6 & 1 & 5 \\ 1 & 7 & 0 \end{bmatrix}$$

14.1 c C*D=
$$\begin{bmatrix} 11 & 16 & 11 \\ 21 & 32 & 21 \\ 22 & 30 & 22 \end{bmatrix}$$

14.1 d 1/B=
$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ -2 & -4 & -5 \end{bmatrix}$$

14.2
```
X= 1.00006
Y= 2.00004
Z= 3
```

```
Tarkka vastaus on

x= 1


X= 1
Y= 2
Z= 3
```

```
Virhe johtuu liukulukumuuttujien
pyöristyksestä ABC 80 :ssa .
```

LIITE 3

Hakemisto

ABC80

ABS 104
ADD 141
aliohjelma 74
alkio 164
alleviivaus 95
apumuuttuja 48
aritmeettinen lauseke 104
ASC 86
ASCII-aritmetiikka 140
ASCII-koodi 86
ASCII-taulukko 88
ATN 104
aurinkomerkki 20
BASIC-kieli 15
CHAIN 65
CHR 87
CLOSE 133
CLRDOT 119
COMP 141
COS 104
CTRL 12
CUR 110
data 15
DATA 22
DEF FN 105
desimaalipiste 35
DIM 51, 173
dimensio 51
DIV 141
DOT 119
ED 34
ehtolohko 68
eksponentti 36
END 77
ERR 22
ERRCODE 160
EXP 104
FIX 104
FOR . . . TO . . . 44
funktio 37, 103
GET 72
GOSUB 75
GOTO 12
graafiset merkit 117
grafiikka 116
grafiikkakartta 126
histogrammi 123
IF . . . THEN . . . 23
IF . . . THEN . . . ELSE . . . 119
indeksoitu muuttuja 51
INPUT 21
INPUT . . . 133
INPUTLINE 89
INPUTLINE . . . 138
INSTR 95
INT 37, 104
jonoalkio 84
jonomerkki 39
järjestelmäkomento 15
kaksoispiste 74
kantaluku 102
kasettinauhuri 62
kenttä 56
kertolasku 35
kokonaislukumuuttuja 39, 140
komento 15
kontrollinäppäin 12
kopiointinäppäin 34
korjailu 34
kursori 4
kuvaruutu 2
käsky 15
käyrän tulostus 120
lainausmerkki 8
lajittelu 47
laskujärjestys 38
lause 15
LEFT 85
LEN 90
LET 20
LIST 10
liukulukumuuttuja 39, 140
LOAD 66
LOAD CAS: 64
LOG 104
looginen operaattori 57
lukumuuttuja 34
matriisi 164

matriisilaskut 164
MERGE 66
MERGE CAS: 65
merkkijono 84
merkkijonomuuttuja 20, 39
merkkiosoitin 12
MID 85
MUL 141
muuttuja 20
muuttujatyypit 39
muuttujien lajittelu 50
nauhuri 62
NEW 11
NEXT 44
NOTRACE 156
NUM 95
numeerinen merkkijono 140
numeromerkki 132
näppäimistö 2
ohjausnäppäin 12
ohjelma 9
ohjelman tallennus 62
ohjelmavirhe 150
ON ERROR GOTO ... 142
ON ... GOSUB ... 142
ON ... GOTO ... 142
ON ... RESTORE 142
OPEN ... ASFILE ... 133
osoitin 4
paluunäppäin 8
PII 104
pilkkumerkki 14
pistekäyrä 109
poistonäppäin 5
potenssiin korotus 102
PREPARE ... ASFILE ... 132
PRINT 5
PRINT ... 132
PRINT CUR 110
prioriteetti 38
prosenttimerkki 39
puolipiste 13
pyöristys 107
pääohjelma 77
rivinumero 9
RND 44, 104
RUN 9
RUN CAS: 64
sarake 164
satunnaisluku 44
SAVE
SETDOT 119
SGN 104
Shift-näppäin 5
silmukka 44
silmukkamuuttuja 44
SIN 104
SPACE 96
SQR 104
STEP 106
STOP 155
STOP LINE ... 13
STRING 95
SUB 141
TAB 109
TAN 104
taulukko 51, 164
taulukon lajittelu 52
tiedon tallennus 130
tiedosto 132
tieto 15
tietolevy 65
tiliote 32
toimintalohko 68
TRACE 153
TV-monitori 2
tyhjennys 11
Upper Case -näppäin 5
VAL 93
valuuttamerkki 20
vektori 51
viive 74
virheilmoitus 22, 159
vuokaavio 68
vuokaaviosymbolit 79
yhtälöryhmä 171
yksikkömatriisi 169
päättelylohko 68

RANDOMIZE 45
READ 22
REM 77
REN 78
RESTORE 146
Return-näppäin 8
RETURN 75
RIGHT 85
rivi 164

Elektroniikan nopea kehitys on johtanut pienten, tehokkaiden tietokoneiden käyttöön aivan uusissa käyttökohteissa ja sovellutusalueilla. Yksityinen kansalainenkin voi nykyään hankkia itselleen tietokoneen lähes väritelevision hinnalla. Näin on syntynyt uusi tietokoneita käyttävä sukupolvi, jonka koulutustarve on huomattava.

Tietokoneen oikea käyttäminen sekä runsaiden valmiiden ohjelmien hyödyntäminen vaatii eräitä perustietoja tietokoneen ohjelmoinnista. Tämä kirja antaa nämä tiedot. Kirja on tarkoitettu käytettäväksi yhdessä ABC80 -mikrotietokoneen kanssa. Opetuksen muodot voivat vaihdella eri yhteyksissä, mutta tätä kirjaa voidaan käyttää niin itseopiskeluun kuin ryhmäopetukseenkin sekä lukiossa ja ammatillisessa opetuksessa että yritysten sisäisessä koulutuksessa ja opintokerhojen järjestämillä kursseilla.

TUOTESELOSTE Kirjan pääasiallinen käyttöalue.

koulumuoto tai ala / koulu-vuosi	10	11	12	13	14
lukio	■	■	■		
kauppa	■	■	■		
tekninen	■	■	■		
ammatilliset kurssikeskukset	■	■			
kansanopistot ja -korkeakoulut	■	■			
kansalais-, työväen- ja vapaaopistot	■	■			
keskiasteen kouluvuosi	1	2	3	4	5

Kirjaa voidaan sove[illegible] sa koulumuodoissa [illegible] dennyskoulutuksessa [illegible]

merkkien selo[illegible] (Otavan e[illegible]

Tekijät
Anders Andersson
Arne Kulbjer
Jan Lundgren
Sören Thornell
Suomentajat:
Fil. maist. Kauko Orra[illegible]
Fil. maist. Esko Peu[illegible]

Tavoite
Tietokoneohjelmoinnin [illegible]
ABC80-mikrotietokonee[illegible]

Lisämateriaali
ABC80-mikrotietokone

Yhteydet
OTAVA, KOULUOSASTO
Uudenmaankatu 10, 00120 Helsinki 12
Puh. 90-647 022

KOTKAN KAUPUNGINKIRJASTO
285 N 0030322

otava K 61
Tilno A328
ISBN 951-1-06080-5